Calhoun
Institutional Archive of the Naval Postgraduate School

Calhoun: The NPS Institutional Archive

DSpace Repository

Theses and Dissertations | 1. Thesis and Dissertation Collection, all items

1978

Theoretical analysis of transonic flow past unstaggered oscillating cascades.

Olsen, Peter Carlton

Monterey, California. Naval Postgraduate School

http://hdl.handle.net/10945/18562

Downloaded from NPS Archive: Calhoun

http://www.nps.edu/library

Calhoun is the Naval Postgraduate School's public access digital repository for research materials and institutional publications created by the NPS community. Calhoun is named for Professor of Mathematics Guy K. Calhoun, NPS's first appointed -- and published -- scholarly author.

Dudley Knox Library / Naval Postgraduate School
411 Dyer Road / 1 University Circle
Monterey, California USA 93943

THEORETICAL ANALYSIS OF TRANSONIC FLOW PAST UNSTAGGERED OSCILLATING CASCADES

Peter Carlton Olsen

NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

THEORETICAL ANALYSIS OF
TRANSONIC FLOW PAST
UNSTAGGERED OSCILLATING CASCADES

by

Peter Carlton Olsen

September 1978

Thesis Advisor: M.F. Platzer

Approved for public release; distribution unlimited.

T185395

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Theoretical Analysis of Transonic Flow Past Unstaggered Oscillating Cascades		5. TYPE OF REPORT & PERIOD COVERED Engineer's Thesis September 1978 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Peter Carlton Olsen		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940		12. REPORT DATE September 1978 13. NUMBER OF PAGES 134
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office) Naval Postgraduate School Monterey, California 93940		15. SECURITY CLASS. (of this report) Unclassified 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Oscillating Airfoil
Collocation Solution
Potential Flow
Transonic Flow

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This paper presents an independent verification of the collocation method as a technique for calculating the lift on an oscillating airfoil in an unstaggered cascade immersed in transonic flow. This method was

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-014-6601

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

(20. ABSTRACT Continued)

originally proposed by Gorelov. Results presented here differ somewhat from those presented by him. Two formulations are shown; one is purely numerical, the second employs an analytic expansion for small frequency.

Approved for public release; distribution unlimited.

Theoretical Analysis of Transonic Flow Past Unstaggered Oscillating Cascades

by

Peter Carlton Olsen
Lieutenant, United States Coast Guard
B.S., United States Coast Guard Academy, 1970
M.S., University of West Florida, 1975
M.S.A.E., Naval Postgraduate School, 1977
M.S.O.R., Naval Postgraduate School, 1978

Submitted in partial fulfillment of the
requirements for the degree of

AERONAUTICAL ENGINEER

from the

NAVAL POSTGRADUATE SCHOOL

September 1978

Thesis
0484
c.1

ABSTRACT

This paper presents an independent verification of the collocation method as a technique for calculating the lift on an oscillating airfoil in an unstaggered cascade immersed in transonic flow. This method was originally proposed by Gorelov. Results presented here differ somewhat from those presented by him. Two formulations are shown; one is purely numerical, the second employs an analytic expansion for small frequency.

TABLE OF CONTENTS

I. INTRODUCTION ---------- 14

II. UNSTEADY TRANSONIC FLOW THEORY ---------- 16

III. SMALL PERTURBATION THEORY OF TRANSONIC FLOW - 23

A. GENERAL CASE ---------- 23

B. BOUNDARY CONDITION ---------- 25

C. NONDIMENSIONALIZATION ---------- 26

D. HARMONIC OSCILLATIONS ---------- 29

IV. LINEARIZATION OF THE GOVERNING EQUATION ----- 31

A. BASIC SOLUTION ---------- 31

B. BOUNDARY CONDITIONS ---------- 34

C. INITIAL CONDITIONS ---------- 34

V. PROBLEM FORMULATION ---------- 36

A. CO-ORDINATE SYSTEM ---------- 36

B. BOUNDARY CONDITIONS ---------- 37

1. Upstream Condition ---------- 37

2. Flow Tangency Condition ---------- 37

C. BASIC SOLUTION TECHNIQUE ---------- 38

D. COLLOCATION SOLUTION OF THE POTENTIAL EQUATION EXPANDED FOR SMALL k ---------- 51

1. Solution for the Unknown Potential Coefficients ---------- 51

2. Calculation of the Potential -------- 57

VI. RESULTS ---------- 63

VII. RECOMMENDATIONS ---------- 70

APPENDIX A. PROGRAM DESCRIPTION ---------- 73

LIST OF REFERENCES ---------- 133

INITIAL DISTRIBUTION LIST ---------- 134

LIST OF SYMBOLS

a	=	local speed of sound	II
a_o	=	speed of sound in the uniform flow	III
c	=	blade semichord	III
f_j	=	elementary function used in collocation solution	V
F	=	specific energy ("head")	II
G	=	function describing the surface of the airfoil as a function of time	III
H	=	function specifying location of blade surface in the vertical axis	III
i	=	$\sqrt{-1}$	II,III,IV,V
k	=	Strouhal number, nondimensional frequency	III,IV,V
m	=	$\sqrt{(\gamma+1)w}$	IV,V
M	=	Mach number = $\frac{\lvert\vec{V}\rvert}{a}$	III
n	=	number of collocation points - 1, order of highest spanning function	V
p	=	pressure	II
	=	nondimensional interblade distance	V
R	=	universal gas constant	II
R.P.	=	"real part of"	III,IV,V
T	=	temperature	II
t	=	time, nondimensional time	II,III,IV,V
U_o	=	uniform velocity from infinity	II,III,VI
u	=	x-component of velocity	II,III
u^o, u^1	=	interference vertical velocities due to reference and adjacent blades respectively, solved so as to satisfy the tangential flow conditions	V,VI

u'	=	small disturbance velocity	III
v	=	y-component of velocity	II,III,IV,V
$\vec{V}$	=	general velocity vector	II
v'	=	small disturbance velocity	III
v^o, v^1	=	vertical velocities due to the reference and adjacent blades respectively, determined from the tangential flow condition	V
w	=	$\tilde{\phi}_x$, a constant used in Gorelov's approximation of the transonic flow potential	IV,V
x	=	horizontal coordinate, may be non-dimensional	
x_*	=	mp (transformed interblade distance in Gorelov's approximation)	V
x_ℓ	=	blade leading edge	IV
x_o	=	center of pitch of the unstaggered cascade	IV,V
y	=	vertical coordinate, may be non-dimensional	
y, y_1	=	vertical coordinates attached to the reference and adjacent blades respectively, may be non-dimensional	IV,V
z, z_1	=	transformed vertical coordinates used in Gorelov's approximation, attached to the reference and adjacent blades respectively. $z = my$, $z_1 = my_1$	IV,V
$\frac{D}{Dt}$	=	substantial derivative w.r.t. time $= \frac{\partial}{\partial t} + \frac{\partial x}{\partial t}\frac{\partial}{\partial x} + \frac{\partial y}{\partial t}\frac{\partial}{\partial y}$ $= \frac{\partial}{\partial t} + u\frac{\partial}{\partial x} + v\frac{\partial}{\partial y}$	II,III
$O(\omega^2)$	=	"of the order of magnitude of ω^2"	V
α	=	angle of attack	II,III,IV,V
α_o	=	maximum amplitude of pitch oscillations	IV

γ	=	ratio of specific heats, c_p/c_v	
δ_{io}	=	Dirac Delta function $= \begin{matrix} 1 & \text{when} & i = 0 \\ 0 & \text{when} & i \neq 0 \end{matrix}$	V
η	=	$\cos^{-1}(-x)$	V
η_*	=	$\cos^{-1}(1-x_*)$	V
$\bar{\eta}$	=	$\cos^{-1}(-s)$	V
$\hat{\eta}$	=	$\cos^{-1}(x_*-x)$	V
θ_j^o, θ_j^1	=	interference potential coefficients for reference and adjacent blades respectively	V
λ	=	k/m^2	IV,V
μ_j^o, μ_j^1	=	Fourier coefficients describing the motion of the reference and adjacent blades respectively	V
ν	=	angular frequency of oscillation	IV,V
ρ	=	density	II
σ	=	phase angle	V
τ	=	cascade solidity, $\frac{2}{p}$	V
Φ	=	general velocity potential	II,III,VI
Φ_o	=	uniform flow velocity potential	
$\tilde{\phi}$	=	steady flow perturbation potential	III,IV
ϕ^o, ϕ^1	=	perturbation potential in collocation solution due to reference and adjacent blades respectively	V,VI
Φ^o, Φ^1	=	tranformed potentials	V
ψ	=	oscillatory flow potential	III,IV,V
Ψ	=	transformed oscillatory potential in Gorelov's coordinates corresponding to ψ	V
ψ^o, ψ^1	=	interference potentials due to reference and adjacent blades respectively	V,VI

Ψ^o, Ψ^1 = transformed potentials in Gorelov's coordinates, coresponding to ψ^o and ψ^1 V

ω = $\frac{k(1-m)^2}{m^4}$

∇ = $\vec{i}\,\frac{\partial}{\partial x} + \vec{j}\,\frac{\partial}{\partial y}$, Gradient operator, $\vec{i}$ and $\vec{j}$ are unit vectors in the x and y direction respectively

Computer Variables

DK = Reduced Frequency, k

DLAMDA = λ

DM2 = m^2

DR = mp = x_*

ETA = η

ETASTR = η_*

IPT = Print Parameter

N = n

NF = not used

OFFSET = r

OMEGA = ω

QALPHA = $0 + i(\lambda - k)$

QCONST = $e^{i\sigma}$

QDCL = $C\ell_\alpha$

QDCM = Cm_α

QDK = $0 + ik$

QEXP = $0 + i\lambda$

QINTAP,QINTRP = Variables used to transmit boundary condition integrals

Q1ABCF,Q1RBCF = Interference coefficients for adjacent and reference blades

Q1COF = Right hand side vector in collocation solution

Q1INT = Known matrix of integrals in collocation solution

Q1RBP,Q1ABP = Not used

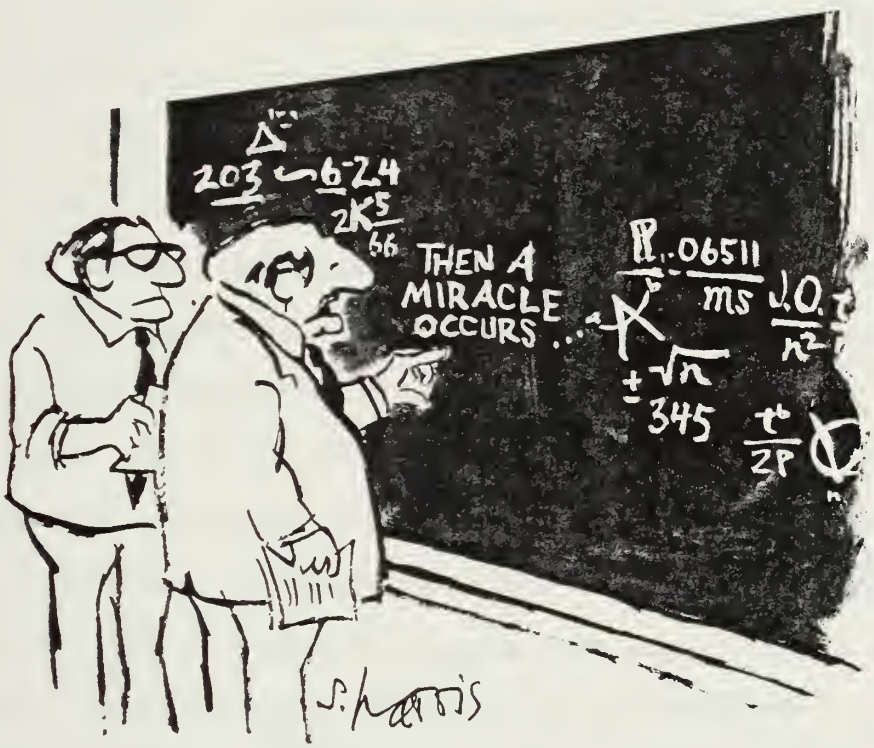
Q2CP = Not used

Q2EXP = $e^{-i\lambda x}$

Q2PT	=	Not used
RHO	=	vertical displacement
RHP	=	local input variable for rho
SIGMA	=	σ
TAU	=	τ
XASTN	=	Current x station in adjacent blade coordinates
XSTN	=	Current x station in reference blade coordinates

ACKNOWLEDGEMENT

I would like to thank Professor Max F. Platzer without whose infinite patience this thesis would not have been written.



"I think you should be more explicit here in step two."

Cartoon Reprinted from American Scientist, Vol. 65, No. 6, Nov-Dec 1977 with permission from Sidney Harris.

I. INTRODUCTION

The analysis of unsteady transonic flows in aircraft turbopropulsion is an area of intense current interest. Rising fuel prices and increasing thrust requirements both point toward the need of turbomachinery capable of performing well with transonic or supersonic internal flow. But, increased flow has increased both the costs and uncertainties of engine designs. Flutter problems have already become a major consideration in engine development. Problems unforeseen in earlier days of turbine engine production have caused long development delays, or forced acceptance of engines producing less than their initial design thrust. These uncertainties cannot be avoided when an attempt is made to extend the state of the art, but they can be reduced by extending the range of analytical modeling.

Such extension must now be done piecemeal. The three-dimensional flows in turbomachinery, including the simultaneous effects of boundary layers, rotation, finite blade thickness, spanwise Mach distributions, and shocks, are well beyond present capability. Perhaps one day complete analysis will be practical, but it is not today. The best that can be done now is to approach the problem from one aspect at a time. Flow through a two dimensional cascade has been a useful tool in this partial analysis.

This thesis was originally to have been an extension of the work of Elder [1] and Schlein [2] to the case of a staggered cascade. Their work, based on Teipel's [3] linearization of the unsteady transonic small perturbation equation, analyzed transonic flow through oscillating unstaggered cascades by use of the collocation method. While the problem was easy to state, it was difficult to solve. Both Elder and Schlein had encountered difficulty in employing the collocation method. Therefore, it was decided that verification of the basic collocation solution presented by Gorelov [4] using a different linearization would be a worthwhile goal in itself.

The following investigation presents a verification of the development in [4], along with numerical results and suggestions for further work.

II. UNSTEADY TRANSONIC FLOW THEORY

Considering inviscid flow only, the following four equations govern the aerodynamic flow problem at hand:

The equation of state

$$p = \rho RT \tag{II-1}$$

and the equations for the conservation of

1. Mass: $\operatorname{div}(\rho\vec{v}) + \frac{\partial \rho}{\partial t} = 0$ (II-2)

2. Momentum: $\frac{D\vec{v}}{Dt} + \frac{1}{\rho}\nabla p = 0$ (II-3)

3. Energy: $\frac{DS}{Dt} = 0$ (II-4)

where

$\vec{v}$ = velocity

p = pressure

S = entropy

R = universal gas constant

T = temperature

t = time

ρ = density

$$\frac{D}{Dt} = \frac{\partial}{\partial t} + \frac{\partial}{\partial x}\frac{\partial x}{\partial t} + \frac{\partial}{\partial y}\frac{\partial y}{\partial t}$$

The analysis starts with a uniform flow from infinity. This flow has velocity U_o parallel to the x-axis. This

formulation can be simplified by working with the total velocity potential, Φ, where

$$u = \frac{\partial \Phi}{\partial x} = \Phi_x = x \text{ component of velocity} = \frac{\partial x}{\partial t} \qquad \text{(II-5)}$$

$$v = \frac{\partial \Phi}{\partial y} = \Phi_y = y \text{ component of velocity} = \frac{\partial y}{\partial t} \qquad \text{(II-6)}$$

Thus, the initial uniform flow is represented by the uniform flow potential

$$\Phi_o = U_o x \qquad \text{(II-7)}$$

This notation may be applied to the conservation equations for mass and momentum. The equation for conservation of mass

$$\text{div}(\rho \vec{v}) + \frac{\partial \rho}{\partial t} = 0 \qquad \text{(II-8)}$$

becomes for two-dimensional unsteady flow

$$\frac{\partial (\rho u)}{\partial x} + \frac{\partial (\rho v)}{\partial y} + \frac{\partial \rho}{\partial t} = 0$$

$$\left[\frac{\partial \rho}{\partial t} + u \frac{\partial \rho}{\partial x} + v \frac{\partial \rho}{\partial y}\right] + \rho \left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}\right) = 0 \qquad \text{(II-9)}$$

but

$$\frac{\partial \rho}{\partial t} + u \frac{\partial \rho}{\partial x} + v \frac{\partial \rho}{\partial y} = \frac{D\rho}{Dt}$$

and

$$u = \Phi_x \quad \text{and} \quad v = \Phi_y$$

Thus

$$\frac{D\rho}{Dt} + \rho(\Phi_{xx} + \Phi_{yy}) = 0$$

and

$$\Phi_{xx} + \Phi_{yy} = -\frac{1}{\rho}\frac{D\rho}{Dt} \tag{II-10}$$

The speed of sound is given by

$$a^2 = \frac{dp}{d\rho}$$

Thus

$$\frac{D\rho}{Dt} = \frac{d\rho}{dp} \cdot \frac{Dp}{Dt} = \frac{1}{a^2}\frac{Dp}{Dt}$$

Applying this to equation (II-10) yields

$$\Phi_{xx} + \Phi_{yy} = -\frac{1}{\rho a^2}\frac{Dp}{Dt} \tag{II-11a}$$

$$\Phi_{xx} + \Phi_{yy} = -\frac{1}{\rho a^2}(up_x + vp_y + p_t) \tag{II-11b}$$

$$= -\frac{1}{\rho a^2}[(\nabla\Phi)\cdot(\nabla p) + p_t] \tag{II-11c}$$

where ∇ is the gradient operator, $P_x = \frac{\partial P}{\partial x}$, $P_y = \frac{\partial P}{\partial y}$, $P_t = \frac{\partial P}{\partial t}$.

Laying this aside for the moment, consider the momentum equation (II-3)

$$\frac{D\vec{v}}{Dt} + \frac{1}{\rho} \nabla p = 0$$

$$\frac{D\vec{v}}{Dt} = \frac{\partial \vec{v}}{\partial t} + (\vec{v} \cdot \nabla) \vec{v}$$

Thus

$$\rho [\frac{\partial \vec{v}}{\partial t} + (\vec{v} \cdot \nabla) \vec{v}] = \nabla p \qquad \text{(II-12)}$$

$$\vec{v} = \nabla \Phi$$

so

$$\frac{\partial \vec{v}}{\partial t} = \frac{\partial}{\partial t}(\nabla \Phi) = \nabla \frac{\partial \Phi}{\partial t} \qquad \text{(II-13)}$$

and

$$(\vec{v} \cdot \nabla) \vec{v} = \nabla \frac{V^2}{2} - \vec{v} \times (\nabla \times \vec{v})$$

where

$$V^2 = u^2 + v^2$$

$$= \vec{v} \cdot \vec{v}$$

For irrotational flow

$$\nabla \times \vec{v} = 0$$

thus

$$(\vec{v} \cdot \nabla)\, \vec{v} = \frac{\nabla V^2}{2} \qquad \text{(II-14)}$$

Thus

$$\frac{\nabla p}{\rho} + \nabla\left[\Phi_t + \frac{V^2}{2}\right] = 0 \qquad \text{(II-15)}$$

which after integration along a streamline becomes

$$\int \frac{dp}{\rho} + \Phi_t + \frac{V^2}{2} = F(t) \qquad \text{(II-16)}$$

For uniform flow from infinity $F(t) = \frac{1}{2} U_o^2$ and thus the final result is

$$\int \frac{dp}{\rho} + \Phi_t + \frac{V^2}{2} = \frac{1}{2} U_o^2 \qquad \text{(II-17)}$$

Differentiation with respect to t gives

$$p_t = -\rho(\Phi_{tt} + \frac{1}{2}\frac{\partial V^2}{\partial t}) \qquad \text{(II-18)}$$

From (II-3)

$$-\nabla p = \rho \frac{D\vec{v}}{Dt} \qquad \text{(II-19)}$$

Substitute (II-18) and (II-19) into (II-11c) to obtain

$$\Phi_{xx} + \Phi_{yy} = \frac{1}{a^2}[(\nabla\Phi) \cdot \frac{D\vec{v}}{Dt} + \Phi_{tt} + \frac{1}{2}\frac{\partial V^2}{\partial t}] \qquad \text{(II-20)}$$

This may be further simplified

$$\frac{D\vec{v}}{Dt} = \frac{\partial\vec{v}}{\partial t} + (\vec{v} \cdot \nabla)\vec{v}$$

$$= \frac{\partial\vec{v}}{\partial t} + \frac{1}{2}\nabla V^2$$

Hence

$$\Phi_{xx} + \Phi_{yy} = \frac{1}{a^2}[(\nabla\Phi)\cdot(\frac{\partial\vec{v}}{\partial t} + \frac{1}{2}\nabla V^2) + \Phi_{tt} + \frac{1}{2}\frac{\partial V^2}{\partial t}] \qquad \text{(II-21)}$$

Expanding terms

$$(\nabla\Phi) \cdot (\frac{\partial\vec{v}}{\partial t}) = \nabla\Phi \cdot (\frac{\partial}{\partial t}\nabla\Phi) = \Phi_x\Phi_{xt} + \Phi_y\Phi_{yt}$$

$$\nabla\Phi \cdot \frac{\nabla V^2}{2} = \frac{\Phi_x{}^2\Phi_{xx}}{2} + \frac{\Phi_y{}^2\Phi_{yy}}{2} + \frac{\Phi_x\Phi_y\Phi_{xy}}{2} + \frac{\Phi_y\Phi_x\Phi_{yx}}{2}$$

$$= \frac{\Phi_x{}^2\Phi_{xx}}{2} + \frac{\Phi_y{}^2\Phi_{yy}}{2} + \frac{2\Phi_x\Phi_y\Phi_{xy}}{2} \quad .$$

$$\frac{1}{2}\frac{\partial V^2}{\partial t} = \frac{1}{2}\frac{\partial}{\partial t}[(\nabla\Phi)\cdot(\nabla\Phi)]$$

$$= \frac{1}{2}[\Phi_x\Phi_{xt} + \Phi_{xt}\Phi_x + \Phi_y\Phi_{yt} + \Phi_{yt}\Phi_y]$$

$$= \Phi_x\Phi_{xt} + \Phi_y\Phi_{yt}$$

The final result obtained by combining terms is

$$\left(1 - \frac{\Phi_x^2}{a^2}\right)\Phi_{xx} + \left(1 - \frac{\Phi_y^2}{a^2}\right)\Phi_{yy} - \frac{2\Phi_x\Phi_y\Phi_{xy}}{a^2}$$

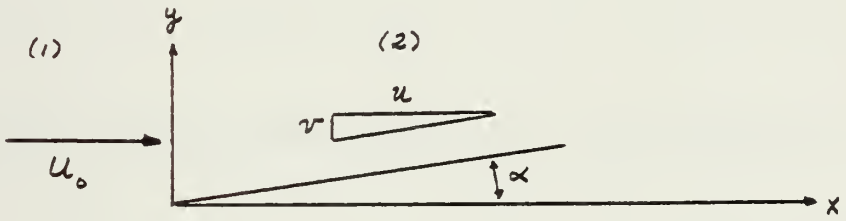
$$- \frac{2\Phi_x}{a^2}\Phi_{xt} - \frac{2\Phi_y}{a^2}\Phi_{yt} - \frac{1}{a^2}\Phi_{tt} = 0 \qquad \text{(II-22)}$$

This result is valid for irrotational, inviscid, two-dimensional, unsteady, compressible flows where gravity has been neglected.

III. SMALL PERTURBATION THEORY OF TRANSONIC FLOW

A. GENERAL CASE

A thin body at a small angle of attack will cause only a slight disturbance in the fluid. A flat plate is an example. Consider flow past a flat plate at angle of attack, α.



The flow at (2) must be parallel to the plate. To achieve this, small disturbance velocities u' and v' must be added to the free stream velocity yielding

$$u = U_o + u'$$

$$v = v'$$

The potential of the disturbed flow may be considered as the sum of the uniform flow potential, $\Phi_o = U_o x$, and the disturbance potential, ϕ

$$\Phi = \Phi_o + \phi \qquad \text{(III-1)}$$

Thus

$$\Phi_x = U_o + u' \qquad \text{(III-2a)}$$

$$\Phi_y = v \qquad \text{(III-2b)}$$

If ϕ is a function of time, then

$$\Phi_t = \phi_t$$

This result may be substituted into (II-22) leading to

$$\left[1 - \frac{(U_o + u')^2}{a^2}\right]\phi_{xx} + \left[1 - \frac{v^2}{a^2}\right]\phi_{yy} - 2\frac{(U_o + u')v}{a^2}\phi_{xy}$$

$$- 2\frac{(U_o + u')}{a^2}\phi_{xt} - \frac{2v}{a^2}\phi_{yt} - \frac{1}{a^2}\phi_{tt} = 0 \qquad \text{(III-3)}$$

This expands to yield

$$\left[1 - \frac{U_o^2 + 2U_o u' + u'^2}{a^2}\right]\phi_{xx} + \left[1 - \frac{v^2}{a^2}\right]\phi_{yy} - 2\frac{(U_o v + u'v)}{a^2}\phi_{xy}$$

$$- 2\frac{(U_o + u')}{a^2}\phi_{xt} - \frac{2v}{a^2}\phi_{yt} - \frac{1}{a^2}\phi_{tt} = 0 \qquad \text{(III-4)}$$

Equation (III-4) may be further simplified as shown by Landahl [3]. All non-linear terms except the $\phi_x\phi_{xx}$ product

term can be neglected yielding the following transonic small disturbance equation

$$[(M^2-1) + (\gamma+1)M^2 \frac{\phi_x}{U_o}]\phi_{xx} - \phi_{yy} + \frac{2M_o}{a_o}\phi_{xt} + \frac{1}{a_o}\phi_{tt} = 0 \qquad \text{(III-5)}$$

where a_o is the velocity of sound in the free-stream, and γ is the ratio of specific heats, and M = Mach number.

B. BOUNDARY CONDITION

The tangential flow condition requires that the flow be tangent to the airfoil surface at each instant of time. This means that no fluid may flow through the surface of the airfoil and is expressed by the condition

$$\frac{DG}{Dt} = 0 \quad \text{on} \quad G(x,y,t) \qquad \text{(III-6)}$$

where

$G(x,y,t)$ describes the surface of the body as a function of time.

For a thin airfoil restricted to small oscillations, this may be written as

$$G = y - H(x,t) \qquad \text{(III-7)}$$

where

$H(x,t)$ is the function describing the position of the airfoil.

$H(x,t)$ can be written for harmonic pitch oscillations as

$$H(x,t) = R.P.[\alpha_o(x-x_o)\, e^{i\nu t}] \qquad \text{(III-8)}$$

where the time-varying angle of attack $\alpha(t)$ is given by

$$\alpha(t) = R.P.[\alpha_o e^{i\nu t}]$$

and α_o = maximum amplitude of pitch oscillation

x_o is the pitch axis

ν is the angular frequency of oscillation

$i = \sqrt{-1}$

R.P. = "real part of"

Inserting (III-8) into the flow tangency condition (III-6) gives, after linearization,

$$\phi_y(x,0) = v(x,0) = \alpha_o[U_o + i\nu(x - x_o)]\, e^{i\nu t} \qquad \text{(III-9)}$$

$$\text{on } y = 0$$

This is a condition for the normal velocity to be prescribed at the airfoil's mean position $y = 0$.

C. NONDIMENSIONALIZATION

The terms in equations (III-5) and (III-9) are dimensional. For the following calculations it is convenient to use non-dimensional quantities. Define non-dimensional time and length to be

$$\bar{x} = \frac{x}{c}$$

$$\bar{y} = \frac{y}{c} \qquad \text{(III-10)}$$

$$\bar{t} = \frac{tU_o}{c}$$

where

U_o = uniform velocity from infinity

c = reference length (blade semichord).

The velocity potential in equation (III-5) may be non-dimensionalized as follows. Let

$$\bar{\phi} = \frac{\phi}{U_o c}$$

Hence:

$$\phi = U_o c \bar{\phi}$$

$$\phi_x = U_o c \bar{\phi}_{\bar{x}} \left(\frac{1}{c}\right)$$

$$= U_o \bar{\phi}_{\bar{x}} \qquad \text{(III-11)}$$

and similarly for the other derivatives in (III-5), yielding

$$[(M^2-1) + (\gamma+1)M^2\bar{\phi}_{\bar{x}}]\phi_{\bar{x}\bar{x}} - \bar{\phi}_{\bar{y}\bar{y}} + 2M^2\bar{\phi}_{\bar{x}\bar{t}} + M^2\bar{\phi}_{\bar{t}\bar{t}} = 0 \qquad \text{(III-12)}$$

This equation is non-dimensional.

The boundary condition given in equation (III-9) may be non-dimensionalized in a similar fashion

$$\phi_y(x,0) = \alpha_o[U_o + i\nu(x - x_o)]\, e^{i\nu t} \qquad \text{(III-9)}$$

Thus

$$U_o c\bar{\phi}_{\bar{y}} = \alpha_o[U_o + ik\frac{U_o}{c}(c\bar{x} - c\bar{x}_o]\, e^{i\nu\bar{t}\frac{c}{U_o}} \qquad \text{(III-13)}$$

where

$$k = \frac{\nu c}{U_o} = \text{Strouhal number or reduced frequency}$$

$$U_o c\bar{\phi}_{\bar{y}}\cdot\frac{1}{c} = \alpha_o U_o[1 + ik(\bar{x} - \bar{x}_o)]\, e^{ik\bar{t}} \qquad \text{(III-14)}$$

Thus

$$\bar{\phi}_{\bar{y}} = \bar{v}(\bar{x},0) = \alpha_o[1 + ik(\bar{x} - \bar{x}_o)]\, e^{ik\bar{t}} \qquad \text{(III-15)}$$

Because the final operations are linear in α_o, set $\alpha_o = 1$, yielding

$$\bar{\phi}_{\bar{y}} = \bar{v}(\bar{x},0) = [1 + ik(\bar{x} - \bar{x}_o)]\, e^{ik\bar{t}} \qquad \text{(III-16)}$$

The overbars denoting nondimensional quantities will be dropped from the remainder of the paper. All further quantities shall be assumed appropriately non-dimensional. This yields the following final equations

$$[(M^2-1)+(\gamma+1)M^2\phi_x]\phi_{xx} - \phi_{yy} + 2M^2\phi_{xt} + M^2\phi_{tt} = 0 \qquad \text{(III-17)}$$

and

$$\phi_y(x,0) = v(x,0) = [1 + ik(x - x_o)]\, e^{ikt} \qquad \text{(III-18)}$$

where

all quantities are nondimensional and

$\alpha_o = 1$

D. HARMONIC OSCILLATIONS

In the case of harmonic oscillations, equation (III-17) may be simplified still further.

Let

$$\phi = \tilde{\phi} + \text{R.P.}[\psi e^{ikt}]$$

where

$\tilde{\phi}$ = non-dimensional steady flow potential

ψ = non-dimensional oscillatory flow potential

R.P. = "real part of"

Equation (III-17) then becomes

$$(1-M^2)\psi_{xx} + \psi_{yy} - M^2(\gamma+1)[\psi_x\psi_{xx} + \tilde{\phi}_x\psi_{xx} + \tilde{\phi}_{xx}\psi_x] + M^2k^2\psi - 2iMk^2\psi_x = 0 \qquad \text{(III-19)}$$

For M close to 1, this is a nonlinear mixed elliptic-hyperbolic partial differential equation with variable coefficients, the exact type depending on $\tilde{\phi}_x$ and $\tilde{\phi}_{xx}$. However, because flutter analysis is primarily concerned with the stability of small perturbations about a steady flow, the oscillatory component may be assumed small compared to the steady flow potential and therefore the product term $\psi_x\psi_{xx}$ may be neglected, yielding,

$$(1-M^2)\psi_{xx} + \psi_{yy} - M^2(\gamma+1)[\tilde{\phi}_x\psi_{xx} + \tilde{\phi}_{xx}\psi_x] = 2iM^2k\psi_x + M^2k^2\psi = 0 \qquad \text{(III-20)}$$

IV. LINEARIZATION OF THE GOVERNING EQUATION

The basic flutter equation, (III-20), is still a non-linear, mixed elliptic-hyperbolic partial differential equation with variable coefficients and difficult to solve. It may yet be further simplified.

A. BASIC SOLUTION

For M = 1, equation (III-20) becomes

$$\psi_{yy} - (\gamma+1)[\tilde{\phi}_x\psi_{xx} + \tilde{\phi}_{xx}\psi_x] - 2ik\psi_x + k^2\psi = 0 \qquad \text{(IV-1)}$$

Now assume

$$\tilde{\phi}_x \approx w = \text{constant} \qquad \text{(IV-2)}$$

$$\tilde{\phi}_{xx} \approx 0$$

throughout the interblade channel. Setting

$$\tilde{\phi}_x(\gamma+1) = w(\gamma+1) = m^2 \qquad \text{(IV-3)}$$

yields

$$m^2\psi_{xx} - \psi_{yy} + 2ik\psi_x - k^2\psi = 0 \qquad \text{(IV-4)}$$

The solution to this equation is found in Garrick and Rubinow [5]

$$\psi(x,y) = -\frac{1}{m}\int_{x_\ell}^{x-my} v(s)J_o[\omega\sqrt{(x-s)^2-(my)^2}]e^{i\lambda(s-x)}\,ds \qquad \text{for } y > 0 \qquad \text{(IV-5a)}$$

and

$$\psi(x,y) = \frac{1}{m}\int_{x_\ell}^{x+my} v(s)J_o[\omega\sqrt{(x-s)^2-(my)^2}]e^{i\lambda(s-x)}\,dx \qquad \text{for } y < 0 \qquad \text{(IV-5b)}$$

where

$$v(x) = \lim_{y\to 0}\frac{\partial}{\partial y}\psi(x,y) \ .$$

$v(x)$ may be obtained directly from the tangential flow boundary condition, and

$$\omega = \frac{k^2(1-m^2)}{m^4}$$

$$\lambda = \frac{k}{m^2}\sqrt{1+m^2} \approx \frac{k}{m^2}$$

(where this paper employs the approximation used by Gorelov [4])

x_ℓ = blade leading edge,

Gorelov [4], has proposed a further simplification.

Set

$$z = my \tag{IV-6a}$$

$$\Psi(x,z) = \psi(x,y)\, e^{i\lambda x} . \tag{IV-6b}$$

Equation (IV-4) then becomes

$$\Psi_{xx} - \Psi_{zz} + \omega^2\Psi = 0$$

with solution

$$\Psi(x,z) = -\frac{1}{m} \int_{x_\ell}^{x-z} v(s) J_o[\omega\sqrt{(x-s)^2-z^2}]\, e^{i\lambda s} ds \tag{IV-7a}$$

$$z > 0$$

$$\Psi(x,z) = \frac{1}{m} \int_{x_\ell}^{x-z} v(s) J_o[\omega\sqrt{(x-s)^2-z^2}]\, e^{i\lambda s} ds \tag{IV-7b}$$

$$z < 0$$

where

$$v(x) = me^{-i\lambda x} \lim_{z\to 0} \Psi_z(x,z)$$

v(x) is obtained from the tangential flow boundary condition.

For a thin body immersed in the flow, the solutions for $y > 0$, $z > 0$, and $y < 0$, $z < 0$ apply above the body along left-running Mach lines, or below along right running Mach lines respectively.

B. BOUNDARY CONDITIONS

1. Flow Tangency Condition

The boundary condition comes from the tangential flow condition, (III-18)

$$v(x) = [1 + ik(x - x_o)] \qquad \text{(IV-8)}$$

2. Upstream Condition

The final linearlized equation is a hyperbolic differential equation with boundary condition

$$\psi(x,y) = 0 \qquad \text{(IV-9)}$$

when

$$x - x_\ell < |my|$$

for the solution shown in equations (IV-5) or

$$\Psi(x,z) = 0$$

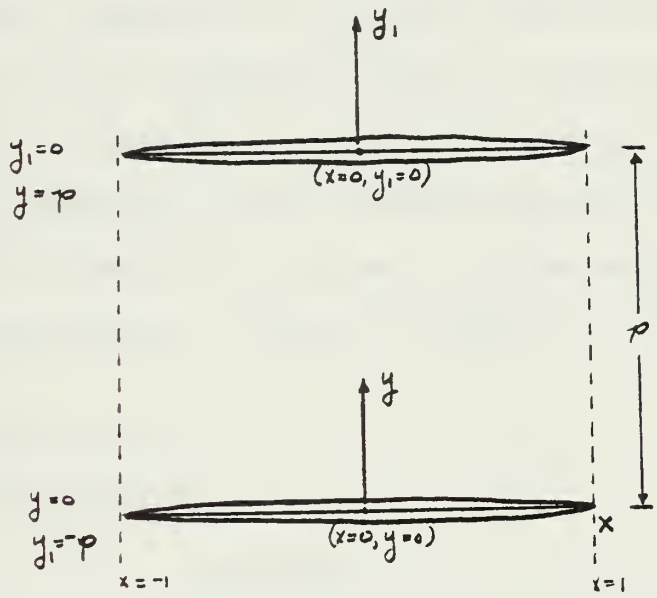
when

$$x - x_{\ell} < |z|$$

for the solution shown in equations (IV-7).

V. PROBLEM FORMULATION

A. CO-ORDINATE SYSTEM



Assume the geometry shown above. Both blades are thin airfoils of semichord c. All measurements are non-dimensional, normalized to c. The (x,y) co-ordinate system has its origin at the center of the reference (lower) blade. The (x,y_1) system is centered at the middle of the adjacent (upper) blade. The origin of the (x,y_1) system is located at (0,p) in the reference system. Generalizing this convention, the same symbols shall be used for the same quantities on both blades. Where discrimination is required, the quantity associated with the adjacent blade will be marked with

superscript ^{1}, the quantity associated with the reference blade will be either unsuperscripted or marked with a superscript ^{o}.

Each blade is assumed to perform a small amplitude harmonic oscillation about its mid-chord point. Both blades are assumed to have identical reduced frequencies, k, and the motion of the adjacent blade lags that of the reference blade by a phase angle σ.

The blades are immersed in a uniform flow from the left at M = 1. The objective is to determine the oscillatory pressure distributions and aerodynamic forces generated by the blades' oscillations. Cascade solidity, $\tau = 2/p$.

B. BOUNDARY CONDITIONS

1. Upstream Condition

$$\psi = 0 \quad \text{whenever}$$

$$x + 1 < |my| \qquad \text{(V-1)}$$

$$\text{and}$$

$$x + 1 < |my_1|, \text{ simultaneously}$$

2. Flow Tangency Condition

Along the reference blade

$$\lim_{y \to 0} \psi_y(x,y) = (1 + ikx) \qquad \text{(V-2a)}$$

Along the adjacent blade

$$\lim_{y_1 \to 0} \psi_{y_1}(x,y_1) = (1 + ikx)e^{i\sigma} \qquad \text{(V-2b)}$$

where

σ is the phase angle between the blades oscillations

C. BASIC SOLUTION TECHNIQUE

Assume that the unsteady potential, ψ, may be written as the sum of four components

$$\psi(x,y) = \phi^o(x,y) + \psi^o(x,y) + \phi^1(x,y_1) + \psi^1(x,y_1) \qquad \text{(V-3)}$$

where:

ϕ^o = potential due to the reference blade alone, known from equation (IV-7)

ϕ^1 = potential due to the adjacent blade alone, known from equation (IV-7)

ψ^o = interference potential required to satisfy tangential flow condition along reference blade, unknown

ψ^1 = interference potential required to satisfy tangential flow condition along adjacent blade, unknown.

This total potential must satisfy the tangential flow condition at the plane of both the reference and adjacent blades. Thus

$$\phi^o_y(x,y=0) + \phi^1_{y_1}(x,y_1=-p) + \psi^o_y(x,y=0) + \psi^1_{y_1}(x,y_1=-p) \qquad \text{(V-4a)}$$

$$= (1 + ikx)$$

at the reference blade, and

$$\phi^{o}(x,y=p) + \phi^{1}_{y_1}(x,y_1=0) + \psi^{o}_{y}(x,y=p) + \psi^{1}_{y_1}(x,y_1=0)$$

$$= (1 + ikx)\, e^{i\sigma} \qquad \text{(V-4b)}$$

at the adjacent blade.

But from the unsteady potential solution for a single oscillating blade one has

$$\phi^{o}_{y}(x,y=0) = 1 + ikx \qquad \text{(V-5a)}$$

and

$$\phi^{1}_{y_1}(x,y_1=0) = (1 + ikx)\, e^{i\sigma} \qquad \text{(V-5b)}$$

Thus

$$\phi^{1}_{y_1}(x,y_1=-p) + \psi^{1}_{y_1}(x,y_1=-p) + \psi^{o}_{y}(x,y=0) = 0 \qquad \text{(V-6a)}$$

along the reference blade, and

$$\phi^{o}_{y}(x,y=p) + \psi^{o}_{y}(x,y=p) + \psi^{1}_{y_1}(x,y_1=0) = 0 \qquad \text{(V-6b)}$$

along the adjacent blade.

From equation (IV-7)

$$\phi^{o}(x,y) = -\frac{1}{m}\int_{-1}^{x-my} v^{o}(s)J_{o}[\omega\sqrt{(x-s)^2-(my)^2}]e^{i\lambda(s-x)}ds$$

$$y > 0 \qquad \text{(V-7a)}$$

$$= \frac{1}{m}\int_{-1}^{x+my} v^{o}(s)J_{o}[\omega\sqrt{(x-s)^2-(my)^2}]e^{i\lambda(s-x)}ds$$

$$y < 0 \qquad \text{(V-7b)}$$

where

$$v^{o}(s) = 1 + iks$$

$$\lambda = k/m^2$$

$$m = (\gamma+1)w$$

$$\omega = \frac{k^2(1-m^2)}{m^4}$$

w = mean value of $\tilde{\phi}_x$ in the channel

$$\phi^{1}(x,y_1) = -\frac{1}{m}\int_{-1}^{x-my_1} v^{1}(s)J_{o}[\omega\sqrt{(x-s)^2-(my_1)^2}]e^{i\lambda(s-x)}ds$$

$$y_1 > 0 \qquad \text{(IV-8a)}$$

$$= \frac{1}{m}\int_{-1}^{x+my_1} v^{1}(s)J_{o}[\omega\sqrt{(x-s)^2-(my_1)^2}]e^{i\lambda(s-x)}ds$$

$$y_1 < 0 \qquad \text{(IV-8b)}$$

where

$$v^1(s) = (1 + iks)e^{i\sigma}$$

Henceforth attention will be restricted to the flow within the channel, $0 \leq y \leq p$, $-p \leq y_1 \leq 0$ leaving (IV-7a) and (IV-8b) as the governing equations of interest.

The two interference potentials are assumed to have forms identical to (IV-7a) and (IV-8b).

Set

$$\psi^o(x,y) = -\frac{1}{m}\int_{-1}^{x-my} u^o(s)J_o[\omega\sqrt{(x-s)^2-(my)^2}]e^{i\lambda(s-x)}ds$$

$$y > 0 \qquad \text{(V-9a)}$$

$$\psi^1(x,y_1) = \frac{1}{m}\int_{-1}^{x+m} u^1(s)J_o[\omega\sqrt{(x-s)^2-(my_1)^2}]e^{i\lambda(s-x)}ds \qquad \text{(V-9b)}$$

$$y_1 < 0$$

where

$u^o(s)$ and $u^1(s)$ are unknown functions to be determined so as to satisfy equations (V-6)

Substitution of (V -7a), (V -8b) and (V -9) into (V -6) yields

$$u^o(x) + \psi^1_{y_1}(x,y_1=-p) + \phi^o_{y_1}(x,y_1=-p) = 0 \qquad \text{(V-10a)}$$

$$u^1(x) + \psi^o_y(x,y=p) + \phi^o_y(x,y=p) = 0 \qquad \text{(V-10b)}$$

Recalling Gorelov's transformation discussed in [4] and shown in equations (IV-6) above, set

$$\Phi^o(x,z) = \phi^o(x,y)\ e^{i\lambda x}$$

$$\Phi^1(x,z_1) = \phi^1(x,y_1)\ e^{i\lambda x}$$

$$\Psi^o(x,z) = \psi^o(x,y)\ e^{i\lambda x}$$

$$\Psi^1(x,z_1) = \psi^1(x,y_1)\ e^{i\lambda x}$$

where $z = my$

$z_1 = my_1$

Then

$$\frac{e^{i\lambda x}}{m} u^o(x) + \Phi^1_{z_1}(x,z_1=-x_*) + \Psi^1_{z_1}(x,z_1=-x_*) = 0 \qquad \text{(V-11a)}$$

and

$$\frac{e^{i\lambda x}}{m} u^1(x) + \Phi^o_z(x,z = x_*) + \Psi^o_z(x,z=x_*) = 0 \qquad \text{(V-11b)}$$

where

$$x_* = mp\ .$$

To employ the collocation method, assume that $u^1(x)$ and $u^o(x)$ can be approximated as the sum of a set of elementary functions f_j so that

$$u^o(x) \simeq \sum_{j=0}^{n} \theta_j^o f_j(x) \qquad \text{(V-12a)}$$

$$u^1(x) \simeq \sum_{j=0}^{n} \theta_j^1 f_j(x) \qquad \text{(V-12b)}$$

where $f_j(x) = 0$ when $x \leq x_* - 1$

Note that here both u^o and u^1 are expressed in terms of the same elementary functions, f_j.

Because of the slightly supersonic nature of the problem observe that $u^o(x) = 0$ and $u^1(x) = 0$ when $x \leq x_* - 1$.

Equations (V-12) may now be rewritten as

$$e^{i\lambda x}\sum\theta_j^o f_j(x) + \frac{\partial}{\partial z_1}\int_{x_*-1}^{x+z_1} \sum\theta_j^1 f_j(s) J_o[\omega\sqrt{(x-s)^2-z_1^2}]e^{i\lambda s}ds$$

$$= -e^{i\sigma}\frac{\partial}{\partial z_1}\int_{-1}^{x+z_1} (1+iks) J_o[\omega\sqrt{(x-s)^2-z_1^2}]e^{i\lambda s}ds$$

$$\text{at} \quad z_1 = -x_* \qquad \text{(V-13a)}$$

$$e^{i\lambda x}\sum\theta_j^1 f_j(x) - \frac{\partial}{\partial z}\int_{x_*-1}^{x-z}\sum\theta_j^o f_j(s)J_o[\omega\sqrt{(x-s)^2-z}]e^{i\lambda s}ds$$

$$= \frac{\partial}{\partial z}\int_{-1}^{x+z}(1+iks)J_o[\omega\sqrt{(x-s)^2-z^2}]e^{i\lambda s}ds$$

$$\text{at} \quad z = x_* \qquad \text{(V-13b)}$$

where

$$f_j(x) = 0 \quad \text{for} \quad x \leq x_* - 1$$

This simplifies to

$$e^{i\lambda x}\sum\theta_j^o f_j(x) + \sum\theta_j^1\{\frac{\partial}{\partial z_1}\int_{x_*-1}^{x+z_1} f_j(s)J_o[\omega\sqrt{(x-s)^2-z_1^2}]e^{i\lambda s}ds\}$$

$$= -e^{i\sigma}\frac{\partial}{\partial z_1}\int_{-1}^{x+z_1}(1+iks)J_o[\omega\sqrt{(x-s)^2-z_1^2}]e^{i\lambda s}ds$$

$$\text{at} \quad z_1 = -x_* \qquad \text{(V-14a)}$$

and

$$e^{i\lambda x}\sum\theta_j^1 f_j(x) - \sum\theta_j^o\{\frac{\partial}{\partial z}\int_{x_*-1}^{x-z} f_j(s)J_o[\omega\sqrt{(x-s)^2-z^2}]e^{i\lambda s}ds$$

$$= \frac{\partial}{\partial z}\int_{-1}^{x-z}(1+iks)J_o[\omega\sqrt{(x-s)^2-z^2}]e^{i\lambda s}ds \qquad \text{(V-14b)}$$

at $z = x_*$ where

$$f_j(x) = 0 \text{ for } x \leq x_*-1.$$

Performing the indicated differentiation yields

$$e^{i\lambda x}\sum\theta_j^0 f_j(x) + \sum\theta_j^1\{\int_{x_*-1}^{x-x_*} f_j(s)\,\frac{J_1[\omega\sqrt{(x-s)^2-x_*^2}]\,\omega x_* e^{i\lambda s}ds}{\sqrt{(x-s)^2-x_*^2}}$$

$$+ f_j(x-x_*)e^{i\lambda(x-x_*)}\}$$

$$= e^{i\sigma}\{-\int_{-1}^{x-x_*}(1+iks)\,\frac{\omega x_* J_1[\omega\sqrt{(x-s)^2-x_*^2}]e^{i\lambda s}ds}{\sqrt{(x-s)^2-x_*^2}}$$

$$- [1+ik(x-x_*)]e^{i\lambda(x-x_*)}\} \qquad \text{(V-15a)}$$

and

$$e^{i\lambda x}\sum\theta_j^1 f_j(x) + \sum\theta_j^0\{\int_{x_*-1}^{x-x_*} f_j(s)\,\frac{J_1[\omega\sqrt{(x-s)^2-x_*^2}]\,\omega x_* e^{i\lambda s}ds}{\sqrt{(x-s)^2-x_*^2}}$$

$$+ f_j(x-x_*)e^{i\lambda(x-x_*)}\}$$

$$= -\int_{-1}^{x-x_*}(1+iks)\frac{\omega x_* J_1[\omega\sqrt{(x-s)^2-x_*^2}e^{i\lambda s}ds}{\sqrt{(x-s)^2-x_*^2}} - [1+ik(x-x_*)]e^{i\lambda(x-x_*)}$$

(V-15b)

Gorelov's formulation, equations [2.6, 2.7, 2.8, and 2.9] of [4], can be obtained directly from equations (V-15) by substituting

$$f_j(x) = \cos j\eta - \cos j\eta_*$$

$$v^o = \sum_{j=0}^{n} \mu_j^o \cos j\eta$$

$$v^1 = \sum_{j=0}^{n} \mu_j^1 \cos j\eta$$

where:

$$\eta = \cos(-x)$$

$$\eta_* = \cos(1-x_*)$$

In comparing the two systems care must be taken to note the differing symbols and coordinate systems. The corresponding quantities are:

Here	in[4]
θ_j^o, θ_j^1	$\nu_{o\sigma}$, $\nu_{i\sigma}$
μ_j^o, μ_j^1	$\theta_{o\sigma}$, $\theta_{i\sigma}$
$\eta = \cos^{-1}(-x)$	$\eta = \cos^{-1}(1-x)$
$\eta_* = \cos^{-1}(1-x_*)$	$\eta_* = \cos^{-1}(1-x_*)$
z, z_1	y, y_1
j	σ
σ	ψ

Here $-1 \leq x \leq 1$; in [4] $0 \leq x \leq 2$. This transformation accounts for the differing definitions of η. Making the substitutions results in the system

$$\sum_{j=0}^{n} \{\theta_j^o[\cos j\eta - (1-\delta_{o1}) \cos j\eta_*]$$

$$+ \theta_j^1 \int_{x_*-1}^{x-x_*} \frac{\partial}{\partial z_1} J_o[\omega\sqrt{(x-s)^2-z_1^2}][\cos j\hat{\eta} - (1-\delta_{io})\cos j\eta_*]e^{i\lambda s}ds$$

$$+ \theta_i^1[\cos j\bar{\eta} - (1-\delta_{io})\cos j\eta_*]e^{i\lambda(x-x_*)}\}$$

$$= -\sum_{j=0}^{n}\{-\mu_j^1 \int_{-1}^{x-x_*} \frac{\partial}{\partial z_1} J_o[\sqrt{(x-s)^2 - z_1^2}] \cos j\hat{\eta}e^{i\lambda s}ds$$

$$- \mu_j^1 \cos j\bar{\eta}e^{i\lambda(x-x_*)}\} \qquad \text{(V-16a)}$$

$$\text{at } z_1 = -x_*$$

$$x > x_*-1$$

and

$$\sum \{\theta_j^1 [\cos j\eta - (1-\delta_{oj}) \cos j\eta_*]$$

$$+ \theta_j^o \int_{x_*-1}^{x-x_*} \frac{\partial}{\partial z} J_o[\omega\sqrt{(x-s)^2-z^2}][\cos j\eta-(1-\delta_{oj})\cos j\eta_*]e^{i\lambda s}ds$$

$$+ \theta_j^o[\cos j\bar{\eta} - (1-\delta_{oj})\cos j\eta_*]e^{i\lambda(x-x_*)}\}$$

$$= \sum_{j=0}^{n} \{\mu_j^o \int_{-1}^{x-x_*} \frac{\partial}{\partial z} J_o[\omega\sqrt{(x-s)^2-z^2}]\cos j\hat{\eta}e^{i\lambda s}ds$$

$$- \mu_j^o \cos j\bar{\eta}e^{i\lambda(x-x_*)}\} \qquad \text{(V-16b)}$$

where:

$$\eta = \arccos(-x)$$

$$\eta_* = \arccos(1-x_*)$$

$$\bar{\eta} = \arccos(-s)$$

$$\hat{\eta} = \arccos(-x+x_*)$$

$$\delta_{\bar{o}j} = \text{Dirac } \delta \text{ function} = \begin{cases} 1 & \text{for } j = 0 \\ 0 & \text{for } j \neq 0 \end{cases}$$

Given the change in coordinates and notation, this system is equivalent to that shown in [4].

This was the system programmed for computer solution. Because the function, $f_j(x)$, is unaffected by the differentiation with respect to y (or z) the exact form used need not be specified, so that the system shown in (V-15) may be programmed with f undetermined. A subroutine may be written to return the function desired and the remaining program left perfectly general. In the program developed with this thesis both the Gorelov functions shown above and the Legendre polynomials were employed. All the integrals may now be evaluated at n+1 points, x_i, on both blades in $mp-1 < x_i < 1$ and the resulting linear system solved for θ_j^o and θ_j^1, $j=1,2,\ldots,n+1$.

The interference potentials may be constructed by taking

$$\psi^o(x,z) = \frac{-1}{m} \int_1^{x-z} [\sum \theta_j^o f_j(x)] J_o[\omega\sqrt{(x-s)^2-z^2}]e^{i\lambda s}ds \qquad z > 0 \qquad \text{(V-17a)}$$

$$\psi^1(x,z_1) = \frac{1}{m} \int_{-1}^{x+z_1} [\sum \theta_j^1 f_j(x)] J_o[\omega\sqrt{(x-s)^2-z_1^2}]e^{i\lambda s}ds \qquad z_1 < 0 \qquad \text{(V-17b)}$$

where

$$f_j(x) = 0, \quad \text{for all} \quad x \leq x_*-1$$

Once the potentials have been calculated as outlined above, the surface pressure may be calculated using the relationship

$$C_p = -2(\psi_x + ik\psi) \qquad \text{(V-18a)}$$

$$= -2[\Psi_x + i(k-\lambda)\Psi]e^{-i\lambda x} \qquad \text{(V-18b)}$$

Because all the plates are assumed to be in steady oscillation with uniform phase shift, σ, between neighboring plates, then

$$v^1(x) = v^o(x)e^{i\sigma}\ , \quad u^1(x) = u^o(x)e^{i\sigma}\ , \quad \Psi(x,y) = -\Psi(x,-y)$$

in this case

$$C_{\ell\alpha} = 2\int_{-1}^{1} [\Psi_x(x,+0) + i(k-\lambda)\Psi(x,+0)]e^{-i\lambda x}dx \qquad \text{(V-19)}$$

where

$$\Psi(x_1,+0) = \frac{-1}{m}\int_{-1}^{x} [(1+iks)+u^o(s)]J_o[\omega(x-s)]e^{i\lambda(s)}ds$$

$$+ \frac{e^{i\sigma}}{m}\int_{-1}^{x-x_*} [(1+iks)+u^o(s)]J_o[\omega\sqrt{(x-s)^2-(x_*)^2}]e^{i\lambda(s)}ds$$

$$\text{where } x_* = mp. \qquad \text{(V-20)}$$

Results from this approach, in the form of values of C_{ℓ_α} for k = 0.1, at various values of w are presented in the results for

approximations based both on Gorelov's formulation, and on the Legendre polynomials.

D. COLLOCATION SOLUTION OF THE POTENTIAL EQUATION EXPANDED FOR SMALL k,

In order to provide a partially independent check of the results of the main program, the Gorelov function representation of the collocation solution was expanded for small k, and solved at two collocation points, n = 2. The resulting potentials, and partial derivatives with respect to x and y were then used to replace the corresponding numerical routines in the main program. The output resulting from the approximations were compared with the purely numerical results obtained from the computer program.

1. Solution For The Unknown Potential Coefficients

The basic system of linear equations used to determine the unknown coefficients is

$$\frac{1}{m} e^{i\lambda x_u{}^o} + \phi^1_{z_1} + \psi^1_{z_1} = 0 \ , \quad z = 0, \quad z_1 = -x_* = -mp \tag{V-21a}$$

$$\frac{1}{m} e^{i\lambda x_u{}^1} + \phi^o_z + \psi^o_z = 0 \ , \quad z_1 = 0, \quad z = x_* = mp \tag{V-21b}$$

where

$$0 = u^o(x) = u^1(x) \quad \text{when} \quad x \leq -1+x_*$$

otherwise

$$u^{o} = \sum_{j=1}^{n} \theta_j^{o}(\cos j\eta - \cos j\eta_*) + \theta_o^{o}$$

$$u^{1} = \sum_{j=1}^{n} \theta_j^{1}(\cos j\eta - \cos j\eta_*) + \theta_o^{1}$$

where

$$\eta = \arccos(-x)$$

$$\eta_* = \arccos(1 - x_*) \ .$$

Thus, for n = 2, the system becomes

$$e^{i\lambda x}\{\theta_o^{o} + \theta_1^{o}(\cos\eta - \cos\eta_*)\}$$

$$+ \frac{\partial}{\partial z_1} \int_{-1}^{x+z_1} v^{1}(s) J_o\left[\omega\sqrt{(x-s)^2 - z_1^2}\right] e^{i\lambda s} ds$$

$$+ \frac{\partial}{\partial z_1} \int_{x_*-1}^{x+z_1} u^{1}(s) J_o\left[\omega\sqrt{(x-s)^2 - z_1^2}\right] e^{i\lambda s} ds = 0$$

$$z_1 = -x_* \qquad \text{(V-22a)}$$

along the reference blade and

$$e^{i\lambda x}\{\theta_o^1 + \theta_1^1(\cos\eta - \cos\eta_*)\} - \frac{\partial}{\partial z}\int_{-1}^{x-z} v^o(s)J_o[\omega\sqrt{(x-s)^2-z^2}]e^{i\lambda s}ds$$

$$- \frac{\partial}{\partial z}\int_{x_*-1}^{x-z} u^o(s)J_o[\omega\sqrt{(x-s)^2 - z^2}]e^{i\lambda s}ds \quad . \qquad \text{(V-22b)}$$

$$z = x_*$$

along the adjacent blade

where $u^o(s) = \theta_o^o + \theta_1^o(\cos\eta - \cos\eta_*)$

$$u^1(s) = \theta_o^1 + \theta_1^1(\cos\eta - \cos\eta_*)$$

$$v^o(s) = 1 + iks$$

$$v^1(s) = (1 + iks)e^{i\sigma}$$

For k sufficiently small, this system may be further simplified by the following approximations

$$J_o[\omega\sqrt{(x-s)^2-z^2}] \approx 1 - 0(\omega^2) \approx 1 \qquad \text{(V-23a)}$$

$$J_o[\omega\sqrt{(x-s)^2-z^2}] \approx 1 - 0(\omega^2) \approx 1 \qquad \text{(V-23b)}$$

$$e^{i\lambda x} \approx 1 + i\lambda x - 0(\lambda^2x^2) \approx 1 + i\lambda x$$

$$e^{i\lambda s} \approx 1 + i\lambda s - 0(\lambda^2s^2) \approx 1 + i\lambda s$$

where $0(\omega^2)$ means "of the order of magnitude of ω^2"

$-1 \leq x \leq 1$, $\quad -1 \leq s \leq x-x_*$

$\lambda = k/m^2$, $\quad \omega^2 = \dfrac{k^2(1-m^2)}{m^4}$

The interference source distributions may be replaced by

$$u^{o}(s) = \theta_{o}^{o} + \theta_{1}^{o}(-s + x_{*}-1)$$

$$u^{1}(s) = \theta_{o}^{1} + \theta_{1}^{1}(-s + x_{*}-1) \ .$$

If higher order terms are neglected, the result is a system linear in k and λ

$$(1+i\lambda x)[\theta_{o}^{o} + \theta_{1}^{o}(-x-1+x_{*})] + \frac{\partial e^{i\sigma}}{\partial z_{1}} \int_{-1}^{x+z_{1}} (1+iks)(1+i\lambda s)ds$$

$$+ \frac{\partial}{\partial z_{1}} \int_{x_{*}-1}^{x+z_{1}} [\theta_{o}^{1} + \theta_{1}^{1}(-s-1+x_{*})](1+i\lambda s)ds = 0$$

$$z_{1} = -x_{*} \qquad \text{(V-24a)}$$

$$(1+i\lambda x)[\theta_{o}^{1} + \theta_{1}^{1}(-x-1+x_{*})] - \frac{\partial}{\partial z} \int_{-1}^{x-z} (1-iks)(1+i\lambda s)ds$$

$$- \frac{\partial}{\partial z} \int_{x_{*}-1}^{x-z} [\theta_{o}^{o} + \theta_{1}^{o}(-s-1+x_{*})](1+i\lambda s)ds = 0$$

$$z = x_{*} \qquad \text{(V-24b)}$$

Product terms containing $(k\lambda) = \frac{k^2}{m^2}$ may be neglected as of higher order in k, yielding

$$(1+i\lambda x)[\theta_o^o + \theta_1^o(-x-1+x_*)] + \frac{\partial e^{i\sigma}}{\partial z_1}\int_{-1}^{x+z_1}[1+i(\lambda+k)s]ds$$

$$+ \frac{\partial}{\partial z_1}\int_{x_*-1}^{x+z_1}\{[\theta_o^1 + \theta_1^1(-s-1+x_*)]$$

$$+ i\lambda s[\theta_o^1 + \theta_1^1(-s-1+x_*)]\}ds = 0$$

$$z_1 = -x_* \qquad \text{(V-25a)}$$

$$(1+i\lambda x)[\theta_o^1 + \theta_1^1(-x-1+x_*)] - \frac{\partial}{\partial z}\int_{-1}^{x-z}[1+i(\lambda+k)s]ds$$

$$- \frac{\partial}{\partial z}\int_{x_*-1}^{x-z}\{[\theta_o^o + \theta_1^o(-s-1+x_*)] + i\lambda s[\theta_1^o + \theta_1^o(-s-1+x_*)]\}ds = 0$$

$$z = x_* \qquad \text{(V-25b)}$$

Evaluating the indicated derivatives yields

$$(1+i\lambda x)[\theta_o^o + \theta_1^o(-x-1+x_*)] + [1+i(k+\lambda)(x-x_*)]e^{i\sigma}$$

$$+ [\theta_o^1 + \theta_1^1(2x_*-x-1)]+i\lambda(x-x_*)[\theta_o^1+\theta_1^1(2x_*-x-1)]\} = 0$$

(V-26a)

$$(1+i\lambda x)[\theta_o^1+\theta_1^1(-x-1+x_*)] + [1+i(\lambda+k)(x-x_*)]$$

$$+ \{[\theta_o^o+\theta_1^o(2x_*-x-1)] + i\lambda(x-x_*)[\theta_o^o+\theta_1^o(2x_*-x-1)]\} = 0$$

(V-26b)

Thus:

$$\theta_o^o(1+i\lambda x) + \theta_1^o[(-x-1+x_*) + i\lambda x(-x-1+x_*)]$$

$$+ \theta_o^1[1\ i\lambda(x-x_*)] + \theta_1^1[(2x_*-x-1) + i\lambda(x-x_*)(2x_*-x-1)]$$

$$= -e^{i\sigma}[1 + i(k+\lambda)(x-x_*)] \qquad \text{(V-27a)}$$

$$\theta_o^1[1 + i\lambda x] + \theta_1^1[(-x+1-x_*) + i\lambda x(-x+1-x_*)]$$

$$+ \theta_o^o[1+i\lambda x(x-x_*)] + \theta_1^o[(2x_*-x-1)+i\lambda(x-x_*)(2x_*-x-1)]$$

$$= -[1 + i(\lambda+k)(x-x_*)] \qquad \text{(V-27b)}$$

This system may be solved at two points, x_1 and x_2, for θ_o^o, θ_1^o, θ_o^1, and θ_1^1.

2. Calculation Of The Potential

The potential is given by

$$\psi(x,y) = \Psi(x,z)e^{-i\lambda x} \tag{V-28}$$

where

$$\Psi(x,z) = -\frac{1}{m}\int_{-1}^{x}[v^{o}(s)+u^{o}(s)]J_{o}[\omega\sqrt{(x-s)^{2}}]e^{i\lambda s}ds$$

$$+\frac{1}{m}\int_{-1}^{x-x_{*}}[v^{1}(s)+u^{1}(s)]J_{1}[\omega\sqrt{(x-s)^{2}-x_{*}^{2}}]e^{i\lambda s}ds.$$

$$u^{o}(s) = u^{1}(s) = 0 \quad \text{for all} \quad s \leq x_{*}-1 .$$

Thus

$$\Psi(x,z) = -\frac{1}{m}\int_{-1}^{x} v^{o}(s)J_{o}[\omega\sqrt{(x-s)^{2}}]e^{i\lambda s}ds$$

$$-\frac{1}{m}\int_{x_{*}-1}^{x} u^{o}(s)J_{o}[\omega\sqrt{(x-s)^{2}}]e^{i\lambda s}ds$$

$$+\frac{1}{m}\int_{-1}^{x-x_{*}} v^{1}(s)J_{o}[\omega\sqrt{(x-s)^{2}-x_{*}^{2}}]e^{i\lambda s}ds$$

$$+\frac{1}{m}\int_{x_{*}-1}^{x-x_{*}} u^{1}(s)J_{o}[\omega\sqrt{(x-s)^{2}-x_{*}^{2}}]e^{i\lambda s}ds \tag{V-29}$$

Making the same small frequency approximations as in the previous section yields

$$\Psi(x,z) = -\frac{1}{m}\int_{-1}^{x} v^{o}(s)(1+i\lambda s)ds$$

$$-\frac{1}{m}\int_{x_*-1}^{x} u^{o}(s)(1+i\lambda s)ds$$

$$+\frac{1}{m}\int_{-1}^{x-x_*} v^{1}(s)(1+i\lambda s)ds$$

$$+\frac{1}{m}\int_{x_*-1}^{x-x_*} u^{1}(s)(1+i\lambda s)ds \qquad \text{(V-30)}$$

From the general formulation

$$\Psi = \Phi^{1} + \Phi^{o} + \Psi^{1} + \Psi^{o} \qquad \text{(V-31)}$$

Thus, along the reference blade

$$-m\Phi^{o}(x,z=0) = \int_{-1}^{x} v^{o}(s)(1+i\lambda s)ds = \int_{-1}^{x} (1+iks)(1+i\lambda s)ds$$

$$= \int_{-1}^{x} [1+i(k+\lambda)s]ds = \left[s+i(k+\lambda)\frac{s^2}{2}\right]_{-1}^{x} \qquad \text{(V-32)}$$

$$= x + i\frac{(k+\lambda)}{2}x + 1 - i\left(\frac{k+\lambda}{2}\right)$$

$$\Phi^{o}(x,z=0) = -\frac{1}{m}[(1+x) + i(\frac{k+\lambda}{2})(x^2-1)] \qquad (V-33)$$

By inspection

$$\Phi^{1}(x,z_1=-x_*) = \frac{e^{i\sigma}}{m}\{(1+x-x_*) + i(\frac{k+\lambda}{2})[(x-x_*)^2-1]\} \qquad (V-34)$$

$$-m\Psi^{o}(x,z=0) = \int_{x_*-1}^{x} u^{o}(s)(1+i\lambda s)ds \qquad (V-35)$$

$$= \int_{x_*-1}^{x} [\theta_o^o + \theta_1^o(-s+x_*-1)](1+i\lambda s)ds$$

$$= \int_{x_*-1}^{x} \theta_o^o(1+i\lambda s) + \theta_1^o(-s+x_*-1)(1+i\lambda s)ds$$

$$= \theta_o^o[(x+1-x_*)+\frac{i\lambda}{2}(x^2-x_*^2+2x_*-1)]$$

$$+ \int_{x_*-1}^{x} \theta_1^o[-x+x_*-1+i\lambda(-s^2+sx_*-s)]ds$$

$$= \theta_o^o[(x+1-x_*)+\frac{i\lambda}{2}(x^2-x_*^2+2x_*-1)]$$

$$+ \theta_1^o\{(-\frac{s^2}{2}+sx_*-s)+i\lambda(-\frac{s^3}{3}+\frac{s^2x_*}{2}-\frac{s^2}{2})]_{x_*-1}^{x}\} \qquad (V-36)$$

$$= \theta_{o}^{o}[(x+1-x_*)+\frac{i\lambda}{2}(x^2-x_*^2+2x_*-1)]$$

$$+ \theta_{1}^{o}\{(-\frac{x^2}{2}+x_*-x)-[-\frac{(x_*-1)^2}{2}+x_*(x_*-1)-(x_*-1)]$$

$$+ i\lambda[(-\frac{x^3}{3}+\frac{x^2x_*}{2}-\frac{x^2}{2})+\frac{(x_*-1)^3}{3}-x_*\frac{(x_*-1)^2}{2}+\frac{(x_*-1)^2}{2}]\}$$

$$= \theta_{o}^{o}[(x+1-x_*)+\frac{i\lambda}{2}(x^2-x_*^2+2x_*-1)]$$

$$+ \theta_{1}^{o}\ -\frac{x^2}{2}+x(x_*-1)+\frac{(x_*-1)^2}{2}-(x_*-1)^2$$

$$+ i\lambda[-\frac{x^3}{3}+x^2\frac{(x_*-1)}{2}+\frac{x_*^3}{3}-x_*^2+x_*-\frac{1}{3}$$

$$-\frac{x_*^3}{2}+x_*^2-\frac{x_*}{2}+\frac{x_*^2}{2}-x_*+\frac{1}{2}]\}$$

$$= \theta_{o}^{o}[(x+1-x_*)+\frac{i\lambda}{2}(x^2-x_*^2+2x_*-1)]$$

$$+ \theta_{1}^{o}\{-\frac{x^2}{2}+x(x_*-1)-\frac{(x_*-1)^2}{2}$$

$$+ i\lambda[-\frac{x^3}{3}+\frac{x^2(x_*-1)}{2}-\frac{x_*^3}{6}+\frac{x_*^2}{2}-\frac{x_*}{2}+\frac{1}{6}]\}$$

$$\psi^{o} = -\frac{1}{m}\{\theta_o^o[(x+1-x_*) + \frac{i\lambda}{2}(x^2 + x_*^2 + 2x_*-1)]$$

$$+ \theta_1^o\{-\frac{x^2}{2} + x(x_*-1) - \frac{(x_*-1)^2}{2}$$

$$+ i\lambda[-\frac{x^3}{3} + \frac{x^2(x_*-1)}{2} - \frac{(x_*-1)^3}{6}]\}\} \qquad \text{(V-37)}$$

$m\psi^1(x,z_1=-x_*)$ may be evaluted by substituting $x-x_*$ for x in the expression for $m\psi^o$ and exchanging θ_o^1 and θ_1^1 for θ_o^o and θ_1^o

$$m\psi^1(x,z_1=-x_*) =$$

$$\theta_o^1\{[(x-x_*)+1-x_*] + \frac{i\lambda}{2}[(x-x_*)^2 - x_*^2 + 2x_* - 1]\}$$

$$+ \theta_1^1\{\frac{-(x-x_*)^2}{2} + (x-x_*)(x_*-1) - \frac{(x_*-1)^2}{2}$$

$$+ i\lambda[-\frac{(x-x_*)^3}{3} + \frac{x^2(x_*-1)}{2} - \frac{(x_*-1)^3}{6}]\} \qquad \text{(V-38)}$$

$$= \theta_o^1(x+1-2x_*) + \frac{i\lambda}{2}[(x^2-2xx_*+x_2^2) - x_*^2 + 2x_* - 1]\}$$

$$+ \theta_1^1\{-\frac{(x^2-2xx_*+x_*^2)}{2} + (x - \frac{3}{2}x_* + \frac{1}{2})(x_*-1)$$

$$+ i\lambda[-\frac{x^3}{3} + x^2x_* - xx_*^2 + \frac{x_*^3}{3} + (x^2-2xx_*+x_*^2)\frac{(x_*-1)}{2}$$

$$- \frac{(x_*-1)^3}{6}]\}$$

$$= \theta_o^1\{(x+1-2x_*) + \frac{i\lambda}{2}[x^2 - 2xx_* + 2x_* - 1]$$

$$+ \theta_1^1\{-\frac{x^2}{2} + xx_* - \frac{x_*^2}{2} + x(x-1) + \frac{1}{2}(1-3x_*)(x_*-1)$$

$$+ i\lambda[-\frac{x^3}{3} + x^2\frac{(3x_*-1)}{2} + x(-\frac{x_*^2-2xx_*^2-1)}{2}$$

$$+ \frac{x_*^3}{3} + \frac{x_*^3}{2} - \frac{1}{2}]\}$$

$$\psi^1(x,z_1=-x_*)$$

$$= \frac{1}{m}\{\theta_o^1\{(x+1-2x_*) + \frac{i\lambda}{2}[x^2 - 2xx_* + 2x_* - 1]$$

$$+ \theta_1^1\{[-\frac{x^2}{2} + x(2x_*-1) - xx_*^2 + 2x_* - \frac{1}{2}]$$

$$+ i\lambda[-\frac{x^3}{3} + x^2\frac{(3x_*-1)}{2} + x\frac{(-3x_*^2-1)}{2} + 5\frac{x_*^3}{6} - \frac{1}{2}]\}\}$$

(V-39)

A comparison of the results for the full program and the approximation is given below for $k = 0.01$, $w = 0.05$, $\sigma = \rho$, $n = 2$, yielding $\omega^2 \approx 6.1 \times 10^{-3}$, $\lambda \approx 0.083$

	Full program	Approx
$x = .1285$		
	ϕ -5.363, .327i	-5.379, .549i
	ϕ_x -8.634, .361i	-8.66, .379i
$x = .5643$		
	ϕ -9.754, .809i	-9.804, 1.000i
	ϕ_x -12.234, .692i	-12.272, .7608i
	C_{ℓ_α} +31.658,-3.1032i	C_{ℓ_α} 31.7019,-3.2165i

VI. RESULTS

The collocation method was used to solve the partial differential equation resulting from the Gorelov approximation of transonic potential flow in an unstaggered cascade. The system was solved using both the spanning functions proposed by Gorelov in [4], resulting in the equations (V-16); and the Legendre polynomials, resulting in equations (V-15) with f_j replaced by the Legendre polynomial, P_j. The resulting values of C_{ℓ_α} for $k = .1$, $\tau = 1$, $\sigma = 1$ and seven collocation points on each blade are presented in figures VI-2, VI-3, and VI-4.

Figure VI -1 presents a diagram which is useful in commenting on the other results. This shows the location of the collocation points and first three interference reflections as a function of w expressed as a percentage of that portion of the chord subject to reflection. The collocation points are equally spaced throughout this interval, 12.5% from the leading edge of the interference zone, 12.5% between each pair of points and 12.5% from the blade trailing edge. The independent variable, w, is plotted vertically so that the dependent variable, percent of chord subject to interference, may be more conveniently visualized along the blade. (The curves are not precisely linear, but are very nearly so in the range shown.)

Figure VI -2 shows the C_{ℓ_α} calculated with $k = 0.0$ in comparison with the results obtained from Ackeret theory.

Agreement is good where there is no reflection and the portion of the blade subject to interference is affected by a constant interference potential, $w \geq 0.11$. Throughout the rest of the curve the results calculated here oscillate above and below the theoretical values. This appears to be due to the discrete nature of the approximation used in the collocation method. Rarely is the fraction of the chord subject to interference reflection equal to the fraction of the collocation points which feel it. Where the collocation point fraction lags, as near $w = 0.6$, the collocation results are lower than those due to Ackeret theory. When the collocation point fraction leads, as it does for $w \leq .04$ and briefly for $w \approx .08$, the the collocation results are higher than those due to Ackeret theory. The fault appears to be an intrinsic feature of the small number of points sampled. This results in a set of coefficients similar to those which would be obtained from a generalized Fourier series based on the integration of the Taylor series expansion about each point. This obviously cannot be a good approximation when both the function and its derivative are discontinuous at the reflections.

Figures VI -3 and VI -4 show the results of using Legendre polynomials and Gorelov's functions as spanning functions. The results for both formulations are identical. Gorelov's results are presented for comparison. Agreement is good for $w > 0.05$ except for an anomalous point, marked A .

It is believed that this anomaly is due to the location of the first reflection just ahead of the last collocation point (cf. "A" on Figure VI -1). This will yield a very small contribution from the reflection potential to the linear system from which the collocation points are determined. The resulting system will have a large dynamic range and may be ill-conditioned.

The discrepancy between these results, and those in [3] for $w < 0.5$ is still unexplained, as is the outlying value for $w = 0.5$.

The discontinuities in the imaginary results are believed to be due primarily to the reflection/collocation interaction explained above.

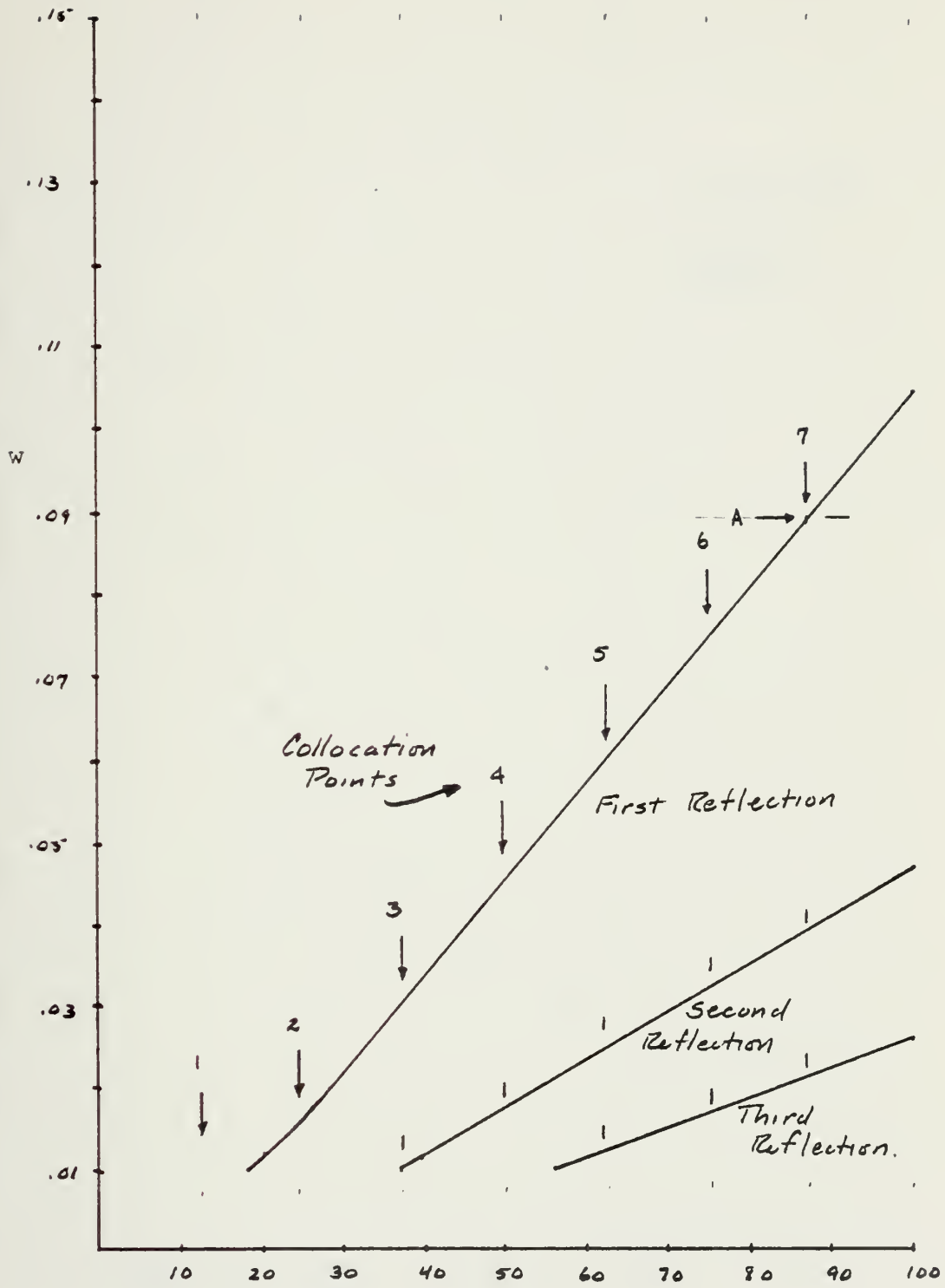


FIGURE VI-1. Location of Reflections and Collocation Points Shown as Percent of Chord Subject to Interference

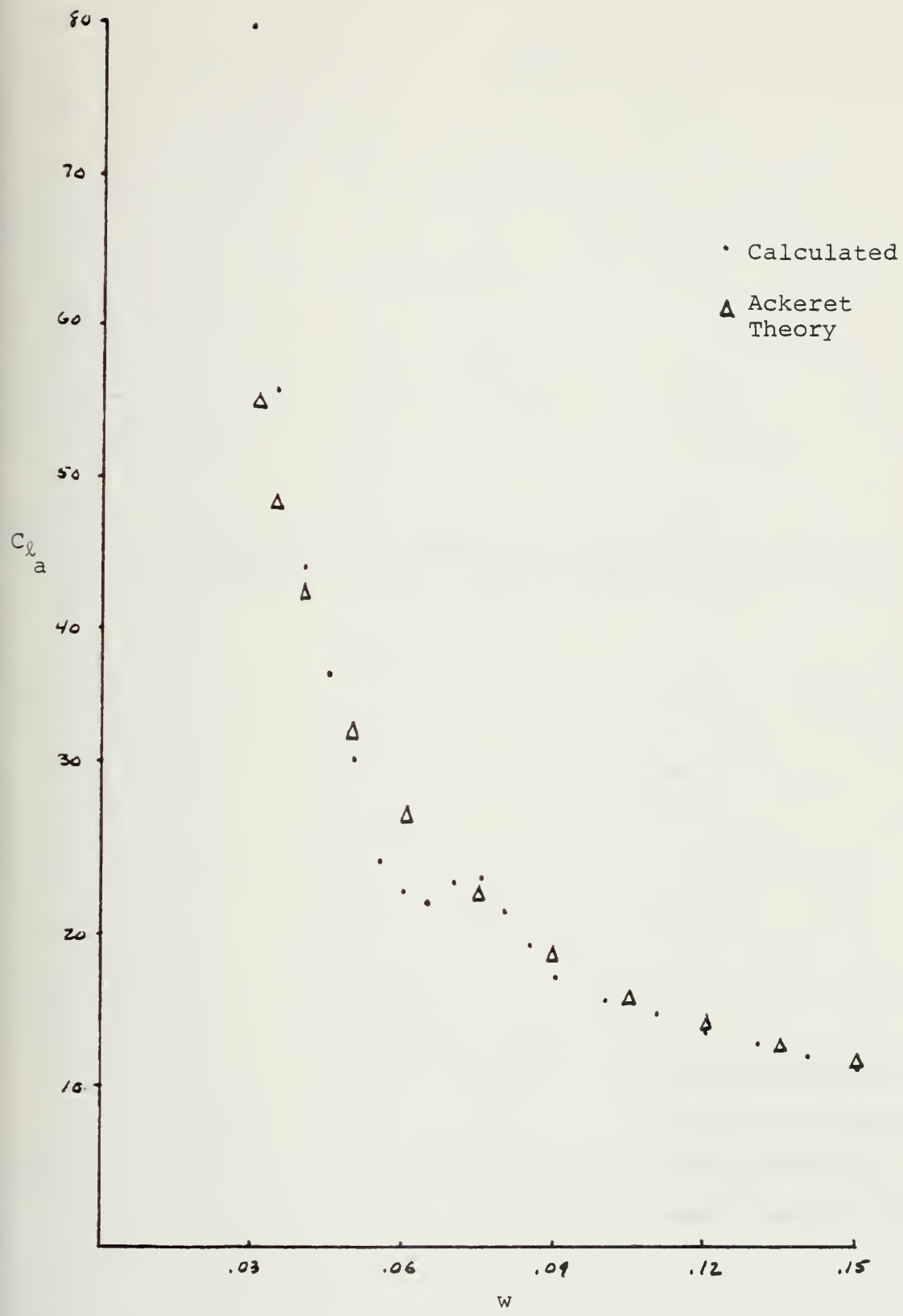


FIGURE VI-2. Comparison of $C\ell_\alpha$-vs-w to that Obtained from Ackeret Theory

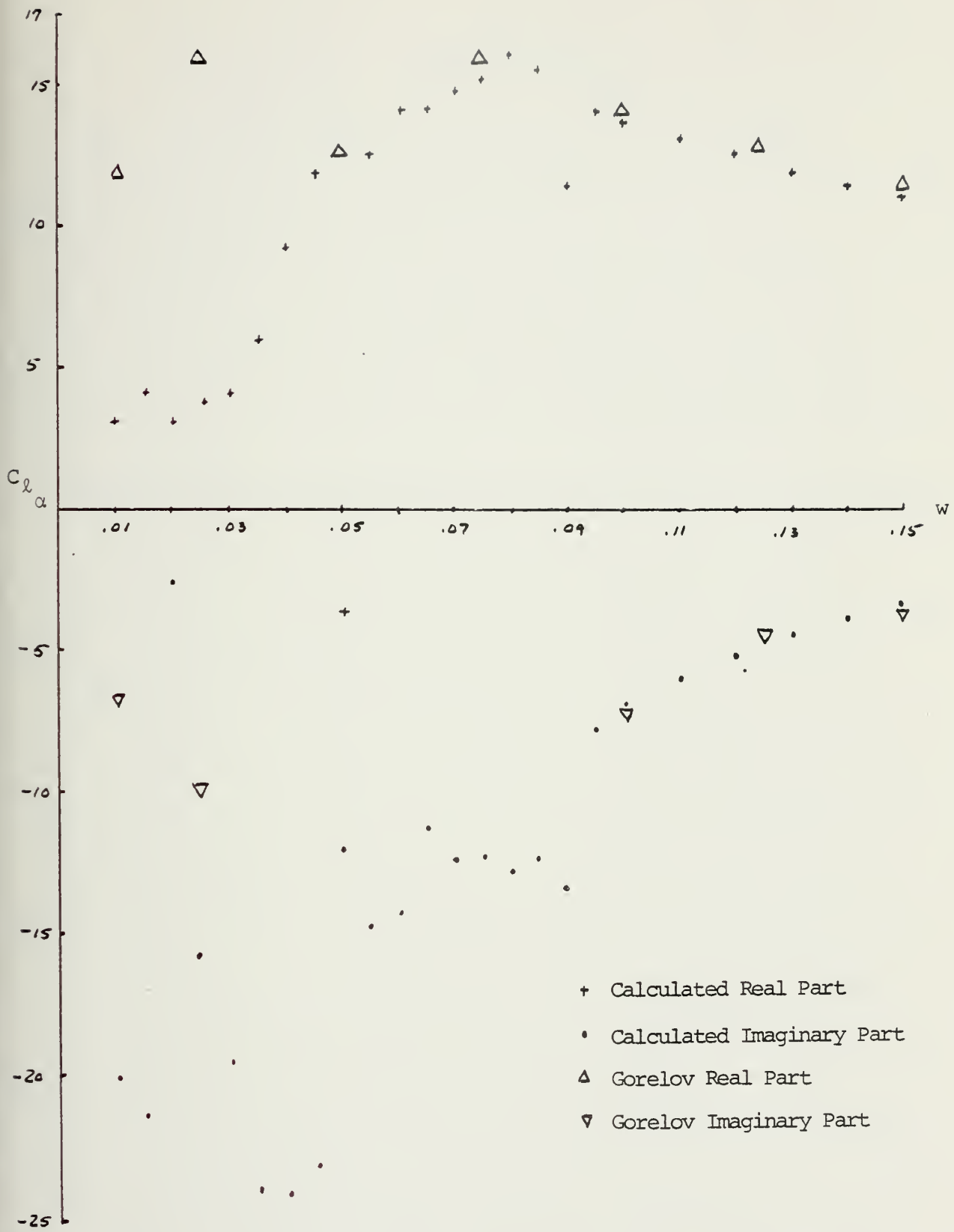


FIGURE VI-3. Plot of $C\ell\alpha$-vs-w, Legendre Polynomials
$k = 0.1$, $\tau = 1.0$, $\sigma = \pi$, $n = 7$
compared with Gorelov's results

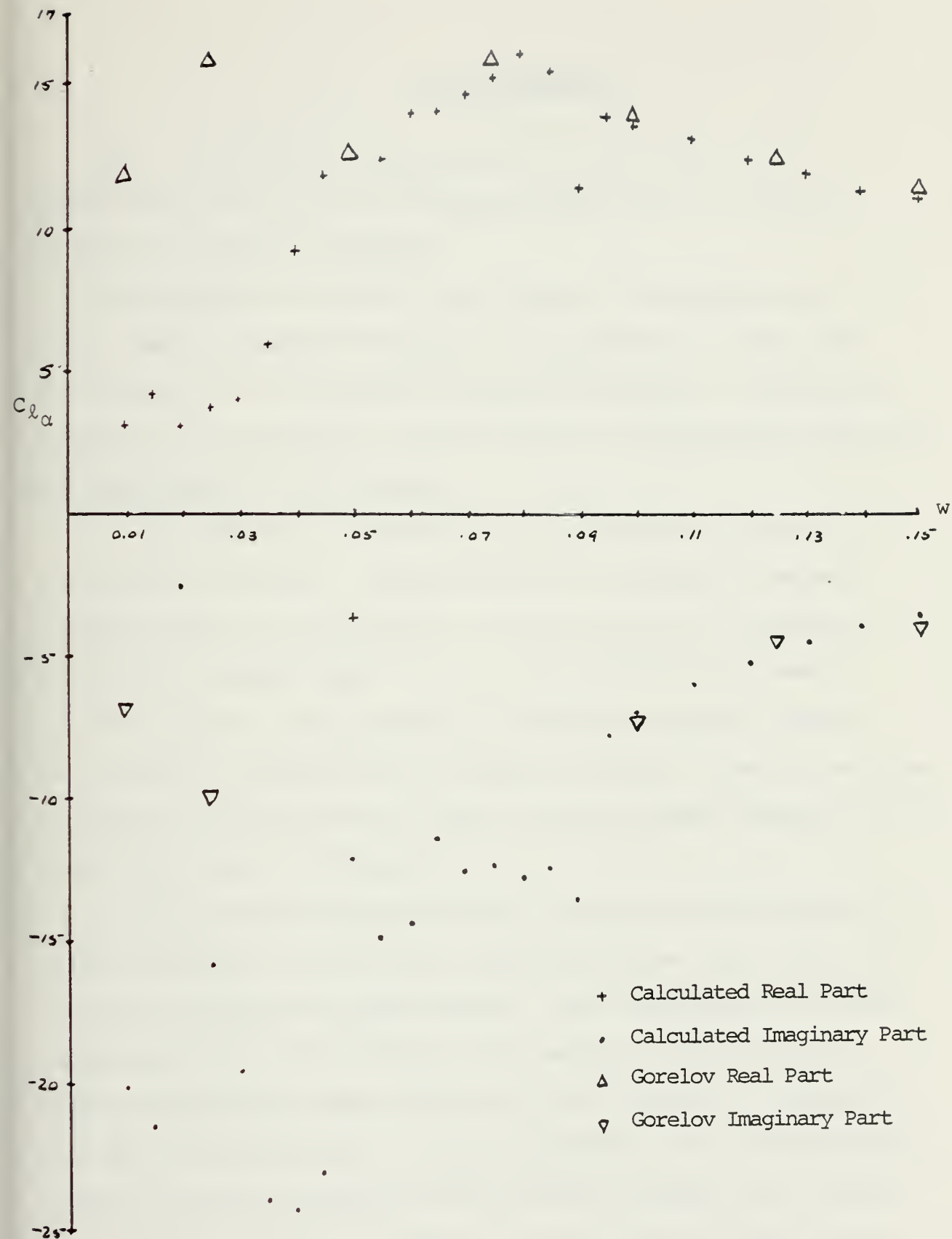


FIGURE VI-4. Plot of $C_{\ell\alpha}$-vs-w, Gorelov Spanning Function
$k = 0.1$, $\tau = 1.0$, $\sigma = \pi$, $n = 7$
compared with Gorelov's results

VII . RECOMMENDATIONS

There are two recommendations to be made about the techniques used in the collocation method, and a new area in which it might be employed.

The program developed in the course of writing this thesis employs adaptive Simpson's integration to calculate the elements of a completely determined system. This system is solved to provide the coefficients of the spanning functions. Two improvements may be made:

1. The Simpson's integration scheme may be replaced by a Gaussian integrator. Experience has shown that several thousand function evaluations are required by the Simpson's integration routine when C_{i_α} is to be evaluated for small w. This entails large amounts of computer time and leads to increased accumulations of numerical error. Use of Gaussian integration would probably improve both of these characteristics with little loss of accuracy.

2. The present program treats a completely determined system of dimension 2n+1 by 2n+1, and then solves that system to find the collocation coefficients. This procedure has worked satisfactorily in this thesis, but may not work as well at higher frequencies where the final linear system of equations may be ill-conditioned. As an alternative, it is recommended that the boundary conditions be applied at more than n points, say m points, where m is twice or three times as many points, and that the least squares technique be used to determine the

the collocation coefficients which give the minimum square error over-all. This may be thought of as "sampling more data" in order to get more information about the unknown function. The present program could be easily modified in this regard by replacing the spanning function matrix, Q1ZINT, by a new matrix of the form

$$Q1ZINT' = X^T X$$

where X is the new m by n+1 (m > n+1) matrix, and replacing the present right-hand-side vector, Q1COF with

$$Q1COF' = X^T Y$$

where Y is the new m by 1 right-hand-side vector. An alternative would be to employ a prepackaged statistical linear regression routine after either modifying the routine to accept complex data, or transforming the present system into a larger system of real numbers only.

The new area in which the collocation method might be employed is the calculation of the potential flow about a staggered cascade. The method coud be employed to calculate both the potential in the channel and above the upper blade. The program presented has been designed to enable the

calculation of flow within the channel of a staggered cascade. Unfortunately, there was not enough time to extend the study to this case.

APPENDIX A

PROGRAM DESCRIPTION

This section describes the computer program used to calculate the interference solution to the Gorelov linearization for unsteady transonic flow in a channel. The program written in IBM Fortran IV with the basic structure outlined by Stevens [5]. The basic points are:

1. Organization of the program into small subroutines, each of which performs a specific task.
2. Transmission data to and from subroutines via a formal parameter argument list. No common statements are used.

The end objective is code which is both easy to modify and maintain.

Each subroutine is designed with optional diagnostic printing of its input and output. This is controlled by the parameter IPT. The diagnostic output is printed (only) if IPT > 0. Each routine accepts IPT, sets IOT = IPT - 1, and then passes IOT as the print parameter to routines it calls. By this method, diagnostic output can be "cascaded" to any desired level. Large initial values of IPT should be avoided because of the spectacular amount of output which can be generated by the double integrals within Q1DCOF.

1. Main Program; including subroutines READ and ABSA.

The basic structure is given in Figure A-1. MAIN calls READ to read input data and then ABSA to calculate the

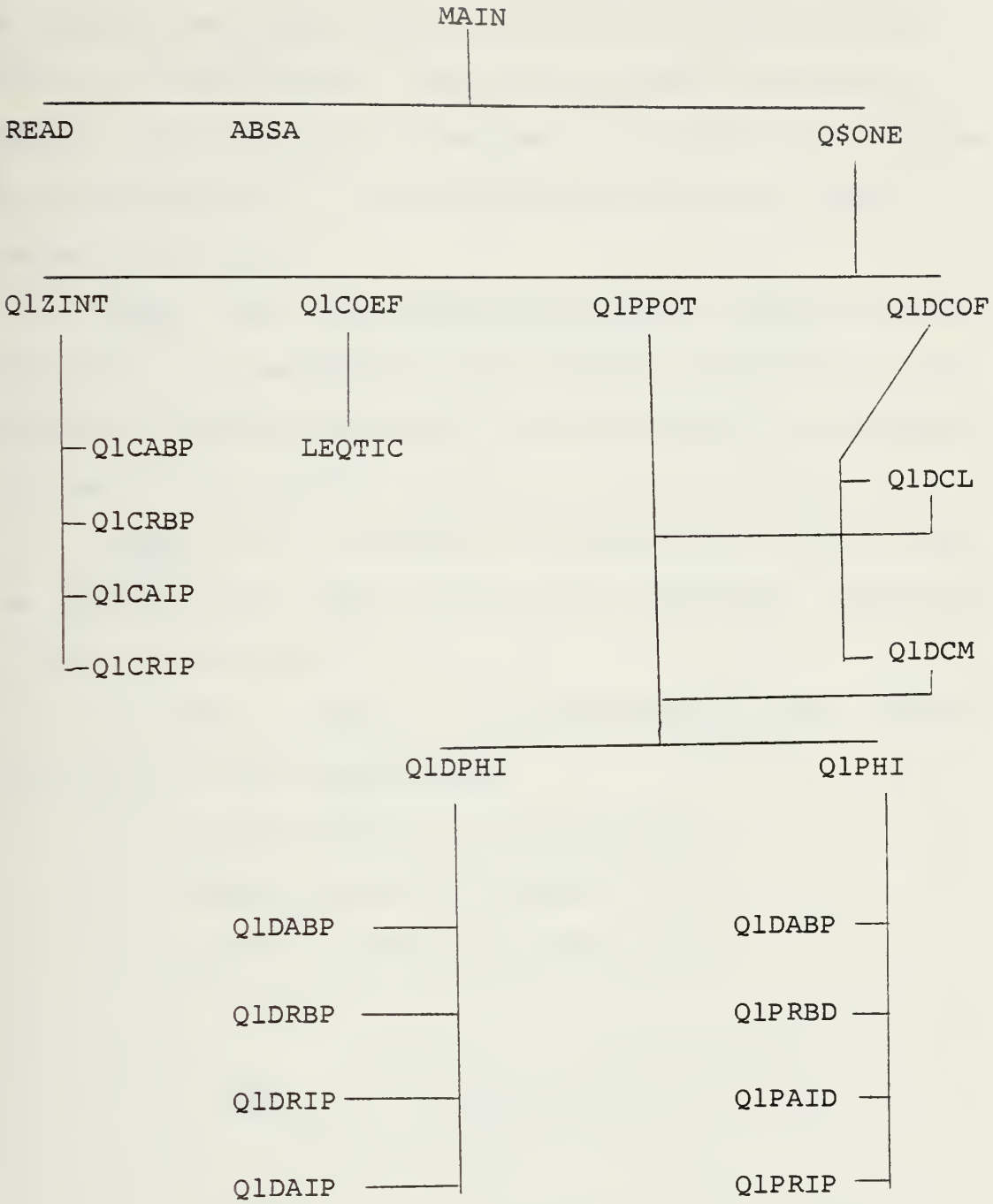


Figure A-1. Program Hierarchy

collocation points. The version of ABSA shown evenly spaces the collocation points across that portion of the blade subject to interference. ABSA may be easily replaced if different point spacing is desired, or if additional points are to be added for an overdetermined system and least squares approximation.

2. Q$ONE This subroutine controls the actual potential calculation. It performs no calculation itself, but calls subordinate subroutines where the calculations are actually performed. The calling hierarchy is shown in Figure A-1.

3. Q1ZINT This subroutine calculates the linear equation system arising from the boundary conditions. Hierarchy is shown in Figure A-2.

The matrix output is carried through Q1INT. Q1ZINT calls the following subprograms

a. Q1CRBP returns the value of ϕ^{o}_{z}

b. Q1CABP returns the value of $\phi^{1}_{z_1}$

c. Q1CRIP returns the values of ψ^{o}_{z}

$$\frac{\partial}{\partial z} \int_{-1+x_*}^{x-z} f_j(s) J_o[\omega\sqrt{(x-s)^2-z^2}]ds$$

where f_i is one of the set of i elementary functions, j=1,n

d. Q1CAIP returns the values of $\psi^{1}_{z_1}$

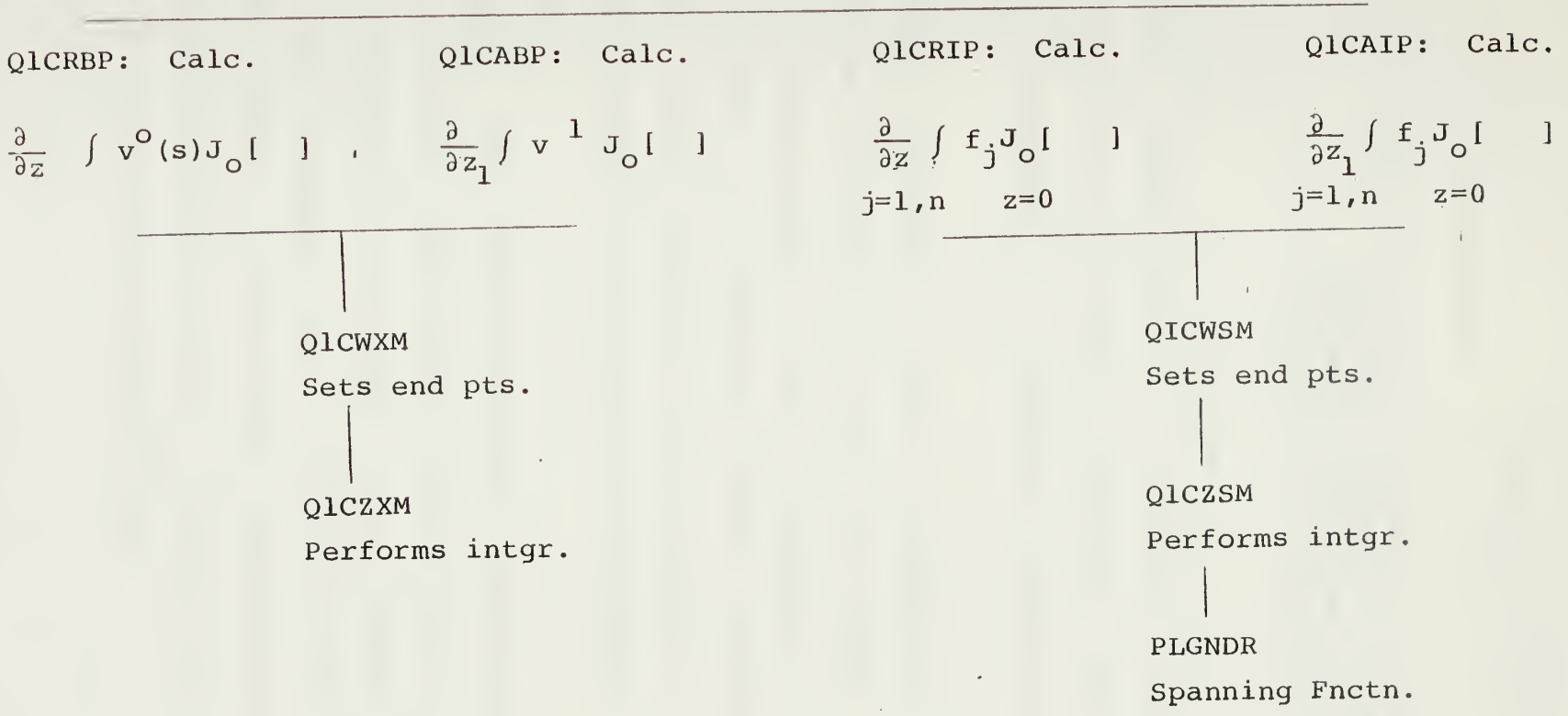

Q1CRBP: Calc.
$\frac{\partial}{\partial z} \int v^{o}(s) J_{o}[\ ]$
Q1CABP: Calc.
$\frac{\partial}{\partial z_1} \int v^{1} J_{o}[\ ]$
Q1CRIP: Calc.
$\frac{\partial}{\partial z} \int f_j J_{o}[\ ]$
j=1,n z=0
Q1CAIP: Calc.
$\frac{\partial}{\partial z_1} \int f_j J_{o}[\ ]$
j=1,n z=0
Q1CWXM
Sets end pts.
Q1CZXM
Performs intgr.
QICWSM
Sets end pts.
Q1CZSM
Performs intgr.
PLGNDR
Spanning Fnctn.

Figure A-2

$$\frac{\partial}{\partial z_1} \int_{-1+OFFSET+x_*}^{x+z_1} f_j(s) J_o[\omega\sqrt{(x-s)^2-y_1^2}]ds$$

where f_j is one of the set of elementary functions. OFFSET is a parameter included to facilitate program conversion to a staggered cascade

e. PLGNDR is the subprogram which returns $f_j(x)$, the elementary spanning function. No other routine contains explicit reference to the spanning function. This facilitates easy replacement of the spanning functions should this be desired.

Q1CRBP and Q1CABP in turn call Q1CWXM and Q1CZXM. Q1CWXM computes end-points and then calls Q1CZXM, a complex integration routine based on SIMP by Shampine and Allen [6]. Q1CAIP and Q1CRIP call Q1CWSM and Q1CZSM to perform the integration. Q1ZINT passes the constant matrix to Q1COEF in the array Q1INT and the right-hand-side vector in the array Q1COF.

4. Q1COEF This subroutine employs the IMSL routine LEQT1C to solve the linear system received from Q1ZINT. The resulting coefficients are Q1ABCF for the adjacent blade and Q1RBCF for the reference blade. LEQT2C, the high precision complex IMSL routine may be directly substituted for LEQT1C. Q1COEF may be rewritten to employ the generalized inverse required for least squares approximation

$$A = (x^T x)^{-1} x^T y \quad \text{where} \quad x = \text{Q1INT}$$

$$y = \text{Q1COF}$$

after first performing the multiplication necessary to replace Q1INT and Q1COF with the proper matrix products in the call to LEQT1C.

5. Q1PPOT This subroutine calculates the potential, ϕ, and ϕ_x, at each collocation point along the reference blade but only if Q1PPOT receives a value of IPT > 0, requiring IPT $\geq$ 2 on input to the main program. If IPT $\leq$ 0, then the subroutine is exited before any calculations are performed. This subroutine is most useful for debugging Q1ZINT and Q1COEF. Q1PPOT calls Q1PABP, Q1PRBP, Q1PRIP, Q1PAIP, Q1DABP, Q1DRBP, Q1DRIP, and Q1DAIP, all of which will be described in the next section.

6. Q1DCOF This subroutine calculates the dimensionless coefficients of lift and moment; C_{ℓ_α}, C_{m_α}. Its internal hierarchy is shown in Figure A-3.

a. Q1DCL calculates the nondimensional complex coefficient C_{ℓ_α}

b. Q1DCM calculates the nondimensional complex coefficient C_{m_α}.

c. Q1PRBP and Q1PABP calculate the potentials due to the reference and adjacent blades respectively. Q1PWXM and Q1PZXM are called to perform the actual integration.

d. Q1PRIP and Q1PAIP calculate the interference potentials along the reference and adjacent blades. Q1PZSM is called to perform the integration.

e. Q1DRBP, Q1DABP, Q1DRIP, and Q1DAIP correspond exactly to subroutines above except that the value returned is the partial derivative of the potential with respect to X. Q1DWXM, Q1DZSM, and Q1DXSM perform the co-reponding integrals.

6. Program Listing The program listing shown below incorporates the Legendre functions as spanning functions. Listings for a subroutine employing Gorelov's spanning function and the linear approximation program follow.

```
      IMPLICIT REAL*8(A-H,O,P,R-Y), COMPLEX*16(Q,Z)                      QLDR0005
      DIMENSION X(13), Q2PT(13), Q2CP(13)                               QLDR0010
C     BLOCK ONE   READ AND EDIT DATA                                    QLDR0015
      CALL ERRSET(208,256,-1,1)                                         QLDR0020
C     WRITE (6,910)                                                     QLDR0025
    1 CALL READ(DK,DR,DW,RHO,OFFSET,SIGMA,N,NF,IPT)                     QLDR0030
C     IF(IPT.GT.0)WRITE(6,990)DK,DR,DW,SIGMA,OFFSET,RHO,N,NF            QLDR0035
  910 FORMAT ('1',5X,'GORELOV CASCADE PROGRAM')                         QLDR0040
  990 FORMAT('0',5X,'INPUT VALUES TRANSMITTED TO MAIN PROGRAM:',/,      QLDR0045
     1' ',10X,'DK',13X,'DR',13X,'DW    ',10X,'SIGMA',10X,'OFFSET',/,  QLDR0050
     1' ',10X,5(E12.5,3X),/,                                            QLDR0055
     2'0',10X,'RHO',12X,'N',5X,'NFCN'                                   QLDR0060
     3,/,' ',10X,1(E12.5,3X), I2,4X,I2)                                 QLDR0065
      CALL ABSA(N,OFFSET,X,RHO,DW    )                                  QLDR0070
C                                                                       QLDR0075
C                                                                       QLDR0080
C     BLOCK TWO    CALLS CALCULATION ROUTINES FOR ZONE ONE AND ZONE TWO  QLDR0085
      CALL Q$ONE(DK,DR,DW,RHO,OFFSET,SIGMA,N,NF,IPT,X)                  QLDR0090
      GO TO 1                                                           QLDR0095
      END                                                               QLDR0100
      SUBROUTINE READ (DK,DR,DW,RHO,OFFSET,SIGMA,N,NF,IPT)              QLDR0105
      IMPLICIT REAL*8(A-H,O,P,R-Y)                                      QLDR0110
      READ(5,905) DK,DR,DW,RHO,OFFSET,SIGMA,N,NF,IPT                    QLDR0115
      IF (DK.LT.0.0D0) STOP                                             QLDR0120
      IF (IPT.GT.0) WRITE (6,910)                                       QLDR0125
      GAMMA = 1.4D0                                                     QLDR0130
      DF = RHO * DSQRT((GAMMA + 1.0D0) * DW)                            QLDR0135
      RHO = DR                                                          QLDR0140
      IF(N.LE.13) GO TO  1                                              QLDR0145
      N = 13                                                            QLDR0150
C     WRITE(6,935 )                                                     QLDR0155
    1 IF(OFFSET.LE.1.9D0) GO TO 2                                       QLDR0160
      WRITE(6,940)                                                      QLDR0165
      OFFSET = OFFSET-1.0D0                                             QLDR0170
      GO TO 1                                                           QLDR0175
    2 CONTINUE                                                          QLDR0180
C     IF (IPT.GT.0) WRITE (6,925) DK,DR,DW,SIGMA,OFFSET,RHO,N,NF,IPT    QLDR0185
      RETURN                                                            QLDR0190
  905 FORMAT(6F10.4,3I2)                                                QLDR0195
  910 FORMAT ('1',10X,'STAGGERED SUPERSONIC CASCADE PROGRAM')           QLDR0200
  925 FORMAT ('0',10X,'DK',13X,'DR',13X,'DW    ',10X,'SIGMA',10X,'OFFSET'QLDR0205
     1,/,' ',10X,5(E12.5,3X),/,                                         QLDR0210
     2'0',10X,'RHO',12X,'N',5X,'NFCN',2X,'IPT'                          QLDR0215
     3,/,' ',10X,1(E12.5,3X), I2,4X,I2,4X,I2)                           QLDR0220
  935 FORMAT('0',5X,'ORIGINAL N TOO LARGE (.GT.13) - RESET TO N = 13')  QLDR0225
  940 FORMAT('0',5X,'OFFSET TOO LARGE (.GT.1.9), RESET AS OFFSET =      QLDR0230
     1OFFSET - 1.0D0')                                                  QLDR0235
      END                                                               QLDR0240
```

```
      SUBROUTINE ABSA (N,OFFSET,X,DR,DW    )                              QLDRC245
      IMPLICIT REAL*8 (A-H,O,P,R-Y), COMPLEX*16(Q,Z)                      QLDRC250
      DIMENSION X(13)                                                     QLDRO255
      XINT = (2.0D0-DR)/DFLOAT(N+1)                                       QLDRC260
      XL = DR - 1.0D0 + 1.0D-8                                            QLDRO265
      DO 10 I = 1,N                                                       QLDRC270
      X(I) = XL + (XINT*DFLOAT(I))                                        QLDRO275
      IF(X(I).EQ.0.0D0) X(I) = .1D-14                                     QLDRC280
   10 CONTINUE                                                            QLDRO285
      RETURN                                                              QLDRC290
      END                                                                 QLDRC295
      SUBROUTINE Q$ONE(DK,DR,DW,RHO,OFFSET,SIGMA,N,NF,IPT,X)              QLDRC300
      IMPLICIT REAL*8(A-H,O,P,R-Y), COMPLEX*16(Z,Q)                       QLDRO305
      DIMENSION Q1COF(26), Q1INT(26,26), Q1RBP(13), Q1ABP(13)             QLDRO310
      DIMENSION Q1RBCF(13), Q1ABCF(13)                                    QLDRO315
      DIMENSION X(13)                                                     QLDRO320
C     IF(IPT.GT.0)WRITE(6,990)DK,DR,RHO,OFFSET,SIGMA,N,NF,IPT             QLDRC325
      ICT = IPT - 1                                                       QLDRO330
  990 FORMAT('0  SUBROUTINE Q$ONE ENTERED WITH:',/,                       QLDRC335
     1' ',5X,'DK',13X,'DR',13X,'RHO',12X,'OFFSET',9X,'SIGMA',10X,         QLDRO340
     2'N',3X,'NF',2X,'IPT',/,                                             QLDRO345
     3  5X,5(E12.5,3X),I2,2X,I2,2X,I3)                                    QLDRC350
      CALL Q1ZINT(DK,DR,DW,RHO,OFFSET,SIGMA,N,X,ICT,Q1COF,Q1INT)          QLDRO355
      CALL Q1COEF(Q1COF,Q1INT,N,IOT,Q1AECF,Q1RBCF)                        QLDRO360
      CALL Q1PPCT(DK,DR,DW,RHO,OFFSET,SIGMA,N,X,Q1AECF,Q1RBCF,ICT)        QLDRC365
      ICT = 0                                                             QLDRO370
      CALL Q1CCCF(DK,DR,DW,RHO,OFFSET,SIGMA,N,Q1AECF,Q1RBCF,ICT)          QLDRC375
      RETURN                                                              QLDRO380
      END                                                                 QLDRC385
C                                                                         QLDRC390
C                                                                         QLDRO395
C                                                                         QLDRO400
      SUBROUTINE Q1ZINT(DK,DR,DW,RHO,OFFSET,SIGMA,N,X,                    QLDRO405
     1      IPT,Q1COF,Q1INT)                                              QLDRO410
      IMPLICIT REAL * 8(A-H,O,P,R-Y), COMPLEX * 16(Z,Q)                   QLDRO415
      DIMENSION Q1COF(26), Q1INT(26,26), Q1NTRP(13), Q1NTAP(13)           QLDRO420
      DIMENSION  X(13)                                                    QLDRC425
  990 FORMAT ('0',10X,'Q1ZINT ENTERED WITH:',/,                           QLDRO430
     1 ' ',10X,'DK',13X,'DR',13X,'RHO',12X,'OFFSET',9X,'SIGMA',10X,       QLDRO435
     2  'N',2X,'IPT',/,                                                  QLDRO440
     3  ' ',10X,5(E12.5,3X),I2,2X,I3)                                    QLDRO445
      CCONST = CDEXP(DCMPLX(0.0D0,SIGMA))                                 QLDRO450
      IF (IPT.GT.0) WRITE (6,990) DK,DR,RHO,OFFSET,SIGMA,N,IPT            QLDRO455
      ICT = IPT - 1                                                       QLDRO460
      GAMMA = 1.4D0                                                       QLDRO465
      CLAMDA = DK/((GAMMA+1.0D0)*DW)                                      QLDRC470
      DO 90 I = 1,N                                                       QLDRO475
      IN = I + N                                                          QLDRC480
```

```
      XSTN = X(I)                                                        QLDR0485
      XO = XSTN - DR                                                     QLDR0490
      CEXP = CDEXP(DCMPLX(0.0D0,DLAMDA*XSTN))                            QLDR0495
      CALL Q1CAIP(DK,DR,DW,RHO,OFFSET,SIGMA,XSTN,CINTAP,N,IOT)           QLDR0500
      CALL Q1CRIP(DK,DR,DW,RHO,OFFSET,XSTN,QINTRP,N,ICT)                 QLDR0505
      DO 20 J = 1,N                                                      QLDR0510
        JN = N + J                                                       QLDR0515
      J1 = J-1                                                           QLDR0520
      Q1INT(I,J)  = PLGNDR(XSTN,DR,J1) * CEXP                            QLDR0525
      C1INT(IN,J) = QINTRP(J)                                            QLDR0530
      Q1INT(I,JN) =  QINTAP(J)                                           QLDR0535
      C1INT(IN,JN) = PLGNDR(XSTN-OFFSET,DR,J1) * CEXP                    QLDR0540
   20 CONTINUE                                                           QLDR0545
      C1COF(IN) = -Q1CRBP(DK,DR,DW,RHO,XSTN,IOT)                         QLDR0550
      Q1COF(I)  = -Q1CABP(DK,DR,DW,RHO,OFFSET,SIGMA,XSTN,IOT)            QLDR0555
   90 CONTINUE                                                           QLDR0560
      RETURN                                                             QLDR0565
      END                                                                QLDR0570
      COMPLEX FUNCTION C1CRBP*16(DK,DR,DW,RHO,XSTN,IPT)                  QLDR0575
      IMPLICIT REAL * 8 (A-H,O,P,R-Y), COMPLEX * 16 (Q,Z)                QLDR0580
      DIMENSION QINP(2)                                                  QLDR0585
      IF (IPT.GT.0) WRITE(6,990) DK,DR,DW,RHO,XSTN,IPT                   QLDR0590
  990 FORMAT('0',10X,'Q1CREP ENTERED WITH:',/,                           QLDR0595
     1   ' ',10X,'DK',11X,'DR',11X,'DW',11X,'RHO',10X,'XSTN',9X,'IPT'    QLDR0600
     2   ,/,' ',10X,5(E12.5,',',),I3)                                   QLDR0605
      IF(XSTN.LE.RHO-1.0D0) GOTO 20                                      QLDR0610
      IF(XSTN.GT.2.0D0) GOTO 20                                          QLDR0615
      ICT = IPT - 1                                                      QLDR0620
      QDK = DCMPLX(0.0D0,DK)                                             QLDR0625
      CALL Q1CWXM(DK,DR,DW,RHO,XSTN,QINP,IOT)                            QLDR0630
      C1CRBP = -QDK*QINP(2) - QINP(1)                                    QLDR0635
      IF(IPT.LE.0) RETURN                                                QLDR0640
      GOTO 30                                                            QLDR0645
   20 Q1CRBP = DCMPLX(0.0D0,0.0D0)                                       QLDR0650
      IF(IPT.LE.0) RETURN                                                QLDR0655
   30 WRITE (6,995) Q1CRBP                                               QLDR0660
  995 FORMAT('0',10X,'C1CRBP = ',E14.7,',',E14.7)                        QLDR0665
      RETURN                                                             QLDR0670
      END                                                                QLDR0675
      COMPLEX FUNCTION Q1CABP*16(DK,DR,DW,RHO,OFFSET,SIGMA,XSTN,IPT)     QLDR0680
      IMPLICIT REAL * 8 (A-H,O,P,R-Y), COMPLEX * 16 (C,Z)                QLDR0685
      DIMENSION QINP(2)                                                  QLDR0690
      IF(IPT.GT.0) WRITE(6,990) DK,DR,DW,RHO,OFFSET,SIGMA,XSTN,IPT       QLDR0695
  990 FORMAT('0',10X,'Q1CABP ENTERED WITH:',/,                           QLDR0700
     1   ' ',10X,'DK',11X,'DR',11X,'DW',11X,'RHO',10X,'OFFSET',7X,       QLDR0705
     2   'SIGMA',8X,'XSTN',9X,'IPT'/,' ',10X,7(E12.5,',',),I3)          QLDR0710
      XASTN = XSTN - OFFSET                                              QLDR0715
      IF(XASTN.LE.RHO-1.0D0) GOTO 20                                     QLDR0720
```

```
      IF(XASTN.GT.2.0D0) GOTO 20                                        QLDR0725
      ICT = IPT - 1                                                     QLDR0730
      CDK = DCMPLX (0.0D0,DK)                                           QLDR0735
      QCONST = CDEXP( DCMPLX(0.0D0,SIGMA))                              QLDR0740
      CALL Q1CWXM(DK,DR,DW,RHO,XASTN,QINP,IOT)                          QLDR0745
      Q1CABP = -((DK*QINP(2) + QINP(1)) * QCONST                        QLDR0750
      IF(IPT.LE.0) RETURN                                               QLDR0755
      GO TO 30                                                          QLDR0760
   20 Q1CABP = DCMPLX(0.0D0,0.0D0)                                      QLDR0765
      IF (IPT.LE.0) RETURN                                              QLDR0770
   30 WRITE (6,995) Q1CABP                                              QLDR0775
  995 FORMAT('0',10X,'Q1CABP = ',E14.7,',',E14.7)                       QLDR0780
      RETURN                                                            QLDR0785
      END                                                               QLDR0790
      SUBROUTINE Q1CWXM(DK,DR,DW,RHO,XSTN,QINP,IPT)                     QLDR0795
C                                                                       QLDR0800
C     N IS THE MAXIMUM DEGREE OF THE TERM U IN THE INTEGRALS            QLDR0805
C                                                                       QLDR0810
      IMPLICIT REAL * 8 (A-H,O,P,R-Y), COMPLEX * 16 (Z,Q)               QLDR0815
      DIMENSION QINP(2)                                                 QLDR0820
      IF(IPT.GT.0) WRITE(6,990) DK,DR,RHO,XSTN,IPT                      QLDR0825
  990 FORMAT('0',10X,'Q1CWXM ENTERED WITH ARGUMENTS:',/,                QLDR0830
     1' ',10X,'DK',16X,'DR',16X,'RHO',15X,'XSTN',14X,'IPT',/,           QLDR0835
     2' ',10X,4(E13.6,5X),I2,5X,I3)                                     QLDR0840
      ICT = IPT - 1                                                     QLDR0845
      B = XSTN - RHO - 1.0D-10                                          QLDR0850
      A = -1.0D0                                                        QLDR0855
      CALL Q1CZXM(DK,DR,DW,RHO,XSTN,A,B,1,QANS,IOT)                     QLDR0860
      QINP(1) = QANS                                                    QLDR0865
      CALL Q1CZXM(DK,DR,DW,RHO,XSTN,A,B,2,QANS,IOT)                     QLDR0870
      QINP(2) = QANS                                                    QLDR0875
      IF(IPT.LE.0) RETURN                                               QLDR0880
      WRITE (6,995)                                                     QLDR0885
  995 FORMAT(' ',10X,'Q1CWXM RESULTS:',/,                               QLDR0890
     1' ',11X,'J',6X,'QINP(I)')                                         QLDR0895
      DO 1001 I = 1,2                                                   QLDR0900
  996 FORMAT(' ',10X,I3,5X,E12.6,',',E12.6)                             QLDR0905
 1001 WRITE (6,996) I, QINP(I)                                          QLDR0910
      RETURN                                                            QLDR0915
      END                                                               QLDR0920
      SUBROUTINE Q1CZXM(DK,DR,DW,RHO,XSTN,A,B,J,QANS,IPT)               QLDR0925
      IMPLICIT REAL*8(A - E,G,H,M,O,P,R - Y), COMPLEX*16(F,G,Z)         QLDR0930
      DIMENSION FV(5),LORR(30),F1T(30),F2T(30),F3T(30),CAT(30),         QLDR0935
     1  ARESTT(30),QESTT(30),EPST(30),QPSUM(30)                         QLDR0940
      F(X) = (X**J1)*CDEXP(QEXP*(X))                                    QLDR0945
     1      *(OMEGA*RHO/(DSQRT((XSTN-X)*(XSTN-X)-YY))*                  QLDR0950
     2      MMBSJ1(OMEGA*DSQRT((XSTN-X)*(XSTN-X)-YY),IER))              QLDR0955
      GAMMA = 1.4D0                                                     QLDR0960
```

```
      DM2 = (GAMMA + 1.CDO) * DW                                                QLDR0965
      CMEGA = DSQRT(DK*CK*(1.0DO-DM2)/(DM2*DM2))                                QLDR0970
      YY = RHO * RHO                                                            QLDRC975
      CLAMDA = DK/DM2                                                           QLDR0980
      QEXP = DCMPLX(0.0DC,CLAMDA)                                               QLDR0985
      J1 = J-1                                                                  QLDR0990
      ACC = 1.0D-6                                                              QLDRC995
      U   =  9.0D-13                                                            QLDR1000
      IF (IPT.GT.C) WRITE(6,990)DK,DR,DW,RHO,XSTN,A,B,J,IFT                     QLDR1005
  990 FORMAT('0',15X,'Q1CZXM ENTERED WITH ARGUMENTS:',/,                        QLDR1010
     1' ',15X,'DK',13X,'DR',13X,'DW    ',10X,'RHO',12X,'XSTN',11X,             QLDR1015
     2'A',14X,'B',14X,'J',2X,'IPT',/,                                           QLDR1020
     3' ',15X,7(E14.7,','),I2,2X,I3)                                            QLDR1025
      EFOURU = 4.0*U                                                            QLDR1030
      IFLAG = 1                                                                 QLDR1035
      EPS = ACC                                                                 QLDR1040
      CERROR = DCMPLX(0.0DO,0.0DO)                                              QLDR1045
      LVL = 1                                                                   QLDR1050
      LCRR(LVL) = 1                                                             QLDR1055
      QPSUM(LVL) = 0.0                                                          QLDR1060
      ALPHA = A                                                                 QLDR1065
      DA = B - A                                                                QLDR1070
      AREA = 0.0                                                                QLDR1075
      ARE ST = 0.0                                                              QLDR1080
      FV(1) = F(ALPHA)                                                          QLDR1085
      FV(3) = F(ALPHA + 0.5*DA)                                                 QLDR1090
      FV(5) = F(ALPHA + DA)                                                     QLDR1095
      KCUNT = 3                                                                 QLDR1100
      WT = DA/6.0                                                               QLDR1105
      QEST = WT*(FV(1) + 4.0*FV(3) + FV(5))                                     QLDR1110
    1 DX = 0.5*DA                                                               QLDR1115
      FV(2) = F(ALPHA + 0.5*DX)                                                 QLDR1120
      FV(4) = F(ALPHA + 1.5*DX)                                                 QLDR1125
      KCLNT = KOUNT + 2                                                         QLDR1130
      WT = DX/6.0                                                               QLDR1135
      QESTL = WT*(FV(1) + 4.0*FV(2) + FV(3))                                    QLDR1140
      QESTR = WT*(FV(3) + 4.0*FV(4) + FV(5))                                    QLDR1145
      QSUM = QESTL + QESTR                                                      QLDR1150
      ARESTL = WT*(CDABS(FV(1)) + CDABS(4.0*FV(2)) + CDABS(FV(3)))              QLDR1155
      ARESTR = WT*(CDABS(FV(3)) + CDABS(4.0*FV(4)) + CDABS(FV(5)))              QLDR1160
      AREA = AREA + ((ARESTL + ARESTR) - AREST)                                 QLDR1165
      QDIFF = QEST - QSUM                                                       QLDR1170
      IF(CDABS(QDIFF).LE.EPS*DABS(AREA))GO TO 2                                 QLDR1175
      IF(DABS(DX).LE.EFOURU*DABS(ALPHA))GO TO 5                                 QLDR1180
      IF(LVL.GE.30)GO TO 5                                                      QLDR1185
      IF(KCUNT.GE.2000)GO TO 6                                                  QLDR1190
      LVL = LVL + 1                                                             QLDR1195
      LCRR(LVL) = 0                                                             QLDR1200
```

```
      F1T(LVL) = FV(3)                                                  QLDR1205
      F2T(LVL) = FV(4)                                                  QLDR1210
      F3T(LVL) = FV(5)                                                  QLDR1215
      DA = DX                                                           QLDR1220
      DAT(LVL) = DX                                                     QLDR1225
      ARES T = ARESTL                                                   QLDR1230
      ARESTT(LVL) = ARESTR                                              QLDR1235
      QEST = QESTL                                                      QLDR1240
      QESTT(LVL) = QESTR                                                QLDR1245
      EPS = EPS/1.4                                                     QLDR1250
      EPST(LVL) = EPS                                                   QLDR1255
      FV(5) = FV(3)                                                     QLDR1260
      FV(3) = FV(2)                                                     QLDR1265
      GO TO 1                                                           QLDR1270
    2 QERROR = QERROR + QDIFF/15.0                                      QLDR1275
    3 IF(LORR(LVL).EQ.0)GO TO 4                                         QLDR1280
      QSUM = QPSUM(LVL) + QSUM                                          QLDR1285
      LVL = LVL - 1                                                     QLDR1290
      IF(LVL.GT.1)GO TO 3                                               QLDR1295
      QANS = QSUM - (B**J1) * CDEXP(QEXPB)                              QLDR1300
      IF(IPT.GT.0) GO TO 11                                             QLDR1305
      IF(IER.EQ.129) GO TO 11                                           QLDR1310
      IF(IFLAG.EQ.1) RETURN                                             QLDR1315
      WRITE(6,990) DK,DR,DW,RHO,XSTN,A,B,J,IPT                          QLDR1320
   11 WRITE(6,995) QANS,IFLAG,IER,QERROR                                QLDR1325
  995 FORMAT(' ',15X,'RESULTS: QANS = ',E14.7,',',E14.7,/,              QLDR1330
     1' ',15X,'IFLAG',2X,'IER',5X,'QERROR',/,                           QLDR1335
     2' ',15X,I3,2X,I3,5X,E14.7,',',E14.7)                              QLDR1340
      RETURN                                                            QLDR1345
    4 QPSUM(LVL) = QSUM                                                 QLDR1350
      LORR(LVL) = 1                                                     QLDR1355
      ALPHA = ALPHA + DA                                                QLDR1360
      DA = DAT(LVL)                                                     QLDR1365
      FV(1) = F1T(LVL)                                                  QLDR1370
      FV(3) = F2T(LVL)                                                  QLDR1375
      FV(5) = F3T(LVL)                                                  QLDR1380
      ARES T = ARESTT(LVL)                                              QLDR1385
      QEST = QESTT(LVL)                                                 QLDR1390
      EPS = EPST(LVL)                                                   QLDR1395
      GO TO 1                                                           QLDR1400
    5 IFLAG = 2                                                         QLDR1405
      GO TO 2                                                           QLDR1410
    6 IFLAG = 3                                                         QLDR1415
      GO TO 2                                                           QLDR1420
      END                                                               QLDR1425
C                                                                       QLDR1430
      SUBROUTINE C1CAIP(DK,DR,DW,RHO,CFST,SGMA,XSTN,CCAIP,N,IPT)        QLDR1435
      IMPLICIT REAL*8 (A-H,O,P,R-Y), COMPLEX * 16 (C,Z)                 QLDR1440
```

```
      DIMENSION  CINP(13), QCAIP(13)                                    QLDR1445
      IF (IPT.GT.0) WRITE(6,990) DK,DR,DW,RHO,OFST,SGMA,XSTN,IPT        QLDR1450
  990 FORMAT('0',10X,'Q1CAIP ENTERED WITH:',/,                          QLDR1455
     1   ' ',10X,'DK',11X,'DR',11X,'DW',11X,'RHO',10X,'OFST',9X,'SGMA',9XQLDR1460
     2   ,'XSTN',9X,'IPT'/,' ',10X,7(E12.5,','),I3)                    QLDR1465
      XASTN = XSTN - OFST                                               QLDR1470
      IF(XASTN.LE.DR+RHO-1.0D0) GO TO 20                                QLDR1475
      IF(XASTN.GT.2.0D0) GOTO 20                                        QLDR1480
      ICT = IPT - 1                                                     QLDR1485
      QCONST = CDEXP( DCMPLX(0.0D0,SGMA))                               QLDR1490
      CALL Q1CWSM(DK,DR,DW,RHO,XASTN,N,CINP,IOT)                        QLDR1495
      DO 10 I = 1,N                                                     QLDR1500
      QCAIP(I) = CINP(I)                                                QLDR1505
   10 CONTINUE                                                          QLDR1510
      IF(IPT.LE.0) RETURN                                               QLDR1515
      GOTO 30                                                           QLDR1520
   20 ZERO = DCMPLX(0.0D0,0.0D0)                                        QLDR1525
      DO 25 I = 1,N                                                     QLDR1530
      QCAIP(I) = ZERO                                                   QLDR1535
   25 CONTINUE                                                          QLDR1540
      IF(IPT.LE.0)RETURN                                                QLDR1545
   30 WRITE (6,995)                                                     QLDR1550
  995 FORMAT ('0',10X,'Q1CAIP RESULTS: J      QCAIP(J)')               QLDR1555
      DO 40 J = 1,N                                                     QLDR1560
      WRITE (6,996) J,QCAIP(J)                                          QLDR1565
  996 FORMAT(' ',26X,I2,3X,E14.7,',',E14.7)                             QLDR1570
   40 CONTINUE                                                          QLDR1575
      RETURN                                                            QLDR1580
      END                                                               QLDR1585
      SUBROUTINE  Q1CRIP (DK,DR,DW,RHO,OFFSET,XSTN,QCRIP,N,IPT)         QLDR1590
      IMPLICIT REAL*8 (A-H,O,P,R-Y), COMPLEX * 16 (Q,Z)                 QLDR1595
      DIMENSION  QCRIP(13)                                              QLDR1600
      IF (IPT.GT.0) WRITE(6,990) DK,DR,DW,RHO,XSTN,IPT                  QLDR1605
  990 FORMAT('0',10X,'Q1CRIP ENTERED WITH:',/,                          QLDR1610
     1   ' ',10X,'DK',11X,'DR',11X,'DW',11X,'RHO',10X,                 QLDR1615
     2   'XSTN',9X,'IPT'/,' ',10X,5(E12.5,','),I3)                     QLDR1620
      IF(XSTN.LE.DR+RHO+OFFSET-1.0D0) GOTO 20                           QLDR1625
      IF(XSTN.GT.2.0D0) GOTO 20                                         QLDR1630
      ICT = IPT - 1                                                     QLDR1635
      CALL Q1CWSM(DK,DR,DW,RHO,XSTN,N,QCRIP,IOT)                        QLDR1640
      IF(IPT.LE.0) RETURN                                               QLDR1645
      GOTO 30                                                           QLDR1650
   20 ZERO = DCMPLX(0.0D0,0.0D0)                                        QLDR1655
      DO 25 I = 1,N                                                     QLDR1660
   25 QCRIP(I) = ZERO                                                   QLDR1665
      IF (IPT.LE.0) RETURN                                              QLDR1670
   30 WRITE (6,995)                                                     QLDR1675
  995 FORMAT ('0',10X,'Q1CRIP RESULTS: J      QCRIP(J)')                QLDR1680
```

```
      DO 40 J = 1,N                                                     QLDR1685
      WRITE (6,996) J,QCRIP(J)                                          QLDR1690
  996 FORMAT(' ',26X,I2,3X,E14.7,',',E14.7)                             QLDR1695
   40 CONTINUE                                                          QLDR1700
      RETURN                                                            QLDR1705
      END                                                               QLDR1710
      SUBROUTINE Q1CWSM(DK,DR,DW,RHO, XSTN,N,QINP,IPT)                  QLDR1715
C                                                                       QLDR1720
C     N IS THE MAXIMUM DEGREE OF THE TERM U IN THE INTEGRALS            QLDR1725
C                                                                       QLDR1730
      IMPLICIT REAL * 8 (A-H,O,P,R-Y), COMPLEX * 16 (Z,Q)               QLDR1735
      DIMENSION QINP(13)                                                QLDR1740
      IF(IPT.GT.0) WRITE(6,990) DK,DR,RHO,XSTN,N,IPT                    QLDR1745
  990 FORMAT('0',10X,'Q1CWSM ENTERED WITH ARGUMENTS:',/,                QLDR1750
     1' ',10X,'DK',16X,'DR',16X,'RHO',15X,'XSTN',14X,'N',6X,'IPT',/,    QLDR1755
     2' ',10X,4(E13.6,5X),I2,5X,I3)                                     QLDR1760
      IOT = IPT - 1                                                     QLDR1765
      B = XSTN - RHO - 1.0D-10                                          QLDR1770
      A = DR-1.0D0                                                      QLDR1775
      DO 30 J = 1,N                                                     QLDR1780
      CALL Q1CZSM(DK,DR,DW,RHO,XSTN,A,B,J,QANS,IOT)                     QLDR1785
      QINP(J) = QANS                                                    QLDR1790
   30 CONTINUE                                                          QLDR1795
      IF(IPT.LE.0) RETURN                                               QLDR1800
      WRITE (6,995)                                                     QLDR1805
  995 FORMAT(' ',10X,'Q1CWSM RESULTS:',/,                               QLDR1810
     1' ',11X,'J',6X,'QINP(I)')                                         QLDR1815
      DO 1001 I = 1,N                                                   QLDR1820
  996 FORMAT(' ',10X,I3,5X,E12.6,',',E12.6)                             QLDR1825
 1001 WRITE (6,996) I, QINP(I)                                          QLDR1830
      RETURN                                                            QLDR1835
      END                                                               QLDR1840
      SUBROUTINE Q1CZSM(DK,DR,DW,RHO,XSTN,A,B,J,QANS,IPT)               QLDR1845
      IMPLICIT REAL*8(A - E,G,H,M,O,P,R - Y), COMPLEX*16(F,Q,Z)         QLDR1850
      DIMENSION FV(5),LORR(30),F1T(30),F2T(30),F3T(30),DAT(30),         QLDR1855
     1  ARESTT(30),QESTT(30),EPST(30),QPSUM(30)                         QLDR1860
      F(X) = PLGNDR(X,DR,J1)*CDEXP(QEXP*(X))                            QLDR1865
     1      *(CMEGA*RHO/(CSQRT((XSTN-X)*(XSTN-X)-YY))*                  QLDR1870
     2      MMBSJ1(OMEGA*DSQRT((XSTN-X)*(XSTN-X)-YY),IER))              QLDR1875
      GAMMA = 1.4D0                                                     QLDR1880
      DM2 = (GAMMA + 1.0D0) * DW                                        QLDR1885
      CMEGA = CSQRT(DK*DK*(1.0D0-DM2)/(DM2*DM2))                        QLDR1890
      YY = RHO * RHO                                                    QLDR1895
      CLAMDA = DK/DM2                                                   QLDR1900
      QEXP = DCMPLX(0.0D0,CLAMDA)                                       QLDR1905
      J1 = J-1                                                          QLDR1910
      ACC = 1.0D-6                                                      QLDR1915
      U   =  9.0D-13                                                    QLDR1920
```

```
      IF (IPT.GT.0) WRITE(6,990)DK,DR,CW,RHO,XSTN,A,B,J,IPT                    QLDR1925
  990 FORMAT('0',15X,'QLCZSM ENTERED WITH ARGUMENTS:',/,                       QLDR1930
     1' ',15X,'DK',13X,'DR',13X,'DW    ',10X,'RHO',12X,'XSTN',11X,            QLDR1935
     2'A',14X,'B',14X,'J',2X,'IPT',/,                                         QLDR1940
     3' ',15X,7(E14.7,','),I2,2X,I3)                                          QLDR1945
      EPSCRU = 4.0*U                                                           QLDR1950
      IFLAG = 1                                                                QLDR1955
      EPS = ACC                                                                QLDR1960
      CERROR = DCMPLX(0.0D0,0.0D0)                                             QLDR1965
      LVL = 1                                                                  QLDR1970
      LORR(LVL) = 1                                                            QLDR1975
      CFSUM(LVL) = 0.0                                                         QLDR1980
      ALPHA = A                                                                QLDR1985
      DA = B - A                                                               QLDR1990
      AREA = C.0                                                               QLDR1995
      AREST = 0.0                                                              QLDR2000
      FV(1) = F(ALPHA)                                                         QLDR2005
      FV(3) = F(ALPHA + 0.5*DA)                                                QLDR2010
      FV(5) = F(ALPHA + DA)                                                    QLDR2015
      KOUNT = 3                                                                QLDR2020
      WT = DA/6.0                                                              QLDR2025
      QEST = WT*(FV(1) + 4.0*FV(3) + FV(5))                                    QLDR2030
    1 DX = 0.5*DA                                                              QLDR2035
      FV(2) = F(ALPHA + 0.5*DX)                                                QLDR2040
      FV(4) = F(ALPHA + 1.5*DX)                                                QLDR2045
      KOUNT = KOUNT + 2                                                        QLDR2050
      WT = DX/6.0                                                              QLDR2055
      QESTL = WT*(FV(1) + 4.0*FV(2) + FV(3))                                   QLDR2060
      QESTR = WT*(FV(3) + 4.0*FV(4) + FV(5))                                   QLDR2065
      QSUM = QESTL + QESTR                                                     QLDR2070
      ARESTL = WT*(CDABS(FV(1)) + CDABS(4.0*FV(2)) + CDABS(FV(3)))             QLDR2075
      ARESTR = WT*(CDABS(FV(3)) + CDABS(4.0*FV(4)) + CDABS(FV(5)))             QLDR2080
      AREA = AREA + ((ARESTL + ARESTR) - AREST)                                QLDR2085
      QDIFF = QEST - QSUM                                                      QLDR2090
      IF(CDABS(QDIFF).LE.EPS*DABS(AREA))GO TO 2                                QLDR2095
      IF(DABS(DX).LE.EPSCRU*DABS(ALPHA))GO TO 5                                QLDR2100
      IF(LVL.GE.30)GO TO 5                                                     QLDR2105
      IF(KOUNT.GE.2000)GO TO 6                                                 QLDR2110
      LVL = LVL + 1                                                            QLDR2115
      LORR(LVL) = 0                                                            QLDR2120
      F1T(LVL) = FV(3)                                                         QLDR2125
      F2T(LVL) = FV(4)                                                         QLDR2130
      F3T(LVL) = FV(5)                                                         QLDR2135
      DA = DX                                                                  QLDR2140
      DAT(LVL) = DX                                                            QLDR2145
      AREST = ARESTL                                                           QLDR2150
      ARESTT(LVL) = ARESTR                                                     QLDR2155
      QEST = QESTL                                                             QLDR2160
```

```
      QESTT(LVL) = QESTR                                                    QLDR2165
      EPS = EPS/1.4                                                         QLDR2170
      EPST(LVL) = EPS                                                       QLDR2175
      FV(5) = FV(3)                                                         QLDR2180
      FV(3) = FV(2)                                                         QLDR2185
      GO TO 1                                                               QLDR2190
    2 QERROR = QERROR + CDIFF/15.0                                          QLDR2195
    3 IF(LCRR(LVL).EQ.0)GO TO 4                                             QLDR2200
      QSUM = QPSUM(LVL) + QSUM                                              QLDR2205
      LVL = LVL - 1                                                         QLDR2210
      IF(LVL.GT.1)GO TO 3                                                   QLDR2215
      CANS = PLGNDR(B,DR,J1) * CDEXP(QEXP*B) - QSUM                         QLDR2220
      IF(IPT.GT.C) GO TO 11                                                 QLDR2225
      IF(IER.EQ.129) GO TO 11                                               QLDR2230
      IF(IFLAG.EQ.1) RETURN                                                 QLDR2235
      WRITE(6,990) DK,DR,DW,RHC,XSTN,A,E,J,IPT                              QLDR2240
   11 WRITE(6,995) QANS,IFLAG,IER,QERRCF                                    QLDR2245
  995 FORMAT(' ',15X,'RESULTS: QANS = ',E14.7,',',E14.7,/,                  QLDR2250
     1' ',15X,'IFLAG',2X,'IER',5X,'QERRCF',/,                               QLDR2255
     2' ',15X,I3,2X,I3,5X,E14.7,',',E14.7)                                  QLDR2260
      RETURN                                                                QLDR2265
    4 QPSUM(LVL) = QSUM                                                     QLDR2270
      LCRR(LVL) = 1                                                         QLDR2275
      ALPHA = ALPHA + DA                                                    QLDR2280
      DA = DAT(LVL)                                                         QLDR2285
      FV(1) = F1T(LVL)                                                      QLDR2290
      FV(3) = F2T(LVL)                                                      QLDR2295
      FV(5) = F3T(LVL)                                                      QLDR2300
      AREST = ARESTT(LVL)                                                   QLDR2305
      QEST = QESTT(LVL)                                                     QLDR2310
      EPS = EPST(LVL)                                                       QLDR2315
      GO TO 1                                                               QLDR2320
    5 IFLAG = 2                                                             QLDR2325
      GO TO 2                                                               QLDR2330
    6 IFLAG = 3                                                             QLDR2335
      GO TO 2                                                               QLDR2340
      END                                                                   QLDR2345
      REAL FUNCTION PLGNDR*8(X,DR,N)                                        QLDR2350
      IMPLICIT REAL*8(A-H,O-Z)                                              QLDR2355
      IF(N.EQ.0)GOTO 100                                                    QLDR2360
      X2 = X*X                                                              QLDR2365
      GOTO (101,102,103,104,105,106,107,108,109,110,111,112), N             QLDR2370
  100 PLGNDR =  1.0D0                                                       QLDR2375
      RETURN                                                                QLDR2380
  101 PLGNDR =  X                                                           QLDR2385
      RETURN                                                                QLDR2390
  102 PLGNDR =  ((3.0D0)*X2-1.0D0)/2.0D0                                    QLDR2395
      RETURN                                                                QLDR2400
```

```
103 PLGNDR =   (5.0D0*X2 - 3.0D0)*X/2.0D0                                       QLDR2405
    RETURN                                                                      QLDR2410
104 PLGNDR =   ((3.5D1*X2 - 3.0D1)*X2 + 3.0D0)/8.0D0                           QLDR2415
    RETURN                                                                      QLDR2420
105 PLGNDR =   ((6.3D1*X2 - 7.0D1) *X2 +1.5D1)*X/8.0D0                         QLDR2425
    RETURN                                                                      QLDR2430
106 PLGNDR=(((231.0D0*X2-315.0D0)*X2+105.0D0)*X2-5.0D0)/16.0D0                 QLDR2435
    RETURN                                                                      QLDR2440
107 PLGNDR=(((429.0D0*X2-693.0D0)*X2+315.0D0)*X2-35.0D0)                       QLDR2445
   1      *X/16.0D0                                                           QLDR2450
    RETURN                                                                      QLDR2455
108 PLGNDR=((((6435.0D0*X2-12012.0D0)*X2+6930.0D0)*X2                          QLDR2460
   1      -1260.0D0)*X2+35.0D0)/128.0D0                                       QLDR2465
    RETURN                                                                      QLDR2470
109 PLGNDR=((((12155.0D0*X2-25740.0D0)*X2+18018.0D0)                           QLDR2475
   1      *X2-4620.0D0)*X2+315.0D0)/128.0D0                                   QLDR2480
    RETURN                                                                      QLDR2485
110 PLGNDR=((((46189.0D0*X2-109395.0D0)*X2+90090.0D0)                          QLDR2490
   1      *X2-30030.0D0)*X2-3465.0D0)*X2-63.0D0)/256.0D0                      QLDR2495
    RETURN                                                                      QLDR2500
111 PLGNDR=((((88179.0D0*X2-230945.0D0)*X2+218790.0D0)*X2                      QLDR2505
   1      -90090.0D0)*X2+15015.0D0)*X2-693.0D0)*X2/256.0D0                    QLDR2510
    RETURN                                                                      QLDR2515
112 PLGNDR=(((((676039.0D0*X2-1939938.0D0)*X2+2078505)*X2                      QLDR2520
   1  -1021020.0D0)*X2+225225.0D0)*X2-18018.0D0)*X2-231.0D0)/1024.0D0          QLDR2525
    RETURN                                                                      QLDR2530
    END                                                                         QLDR2535
    SUBROUTINE Q1COEF( Q1COF,Q1INT,N,IPT,Q1ABCF,Q1RBCF)                         QLDR2540
    IMPLICIT REAL * 8(A-H,O,P,R-Y), COMPLEX * 16 (Z,C)                          QLDR2545
    DIMENSION Q1COF(26), Q1INT(26,26), ZWA(300)                                 QLDR2550
    DIMENSION Q1ABCF(13),Q1RBCF(13)                                             QLDR2555
    IF (IPT.GE.0) WRITE (6,90)                                                  QLDR2560
    IE = 26                                                                     QLDR2565
    M = 1                                                                       QLDR2570
    N2 = 2*N                                                                    QLDR2575
    IF (IPT.LE.0) GO TO 5                                                       QLDR2580
    WRITE (6,98) N,N2                                                           QLDR2585
 98 FORMAT('0',5X,'Q1COEF ENTERED WITH ',I2,' DEG  PWR SERIES (',I2,            QLDR2590
   1 ' SQUARE MATRIX)')                                                         QLDR2595
    DO 2 I = 1,N2                                                               QLDR2600
    WRITE(6,92) I, I, Q1COF(I)                                                  QLDR2605
 92 FORMAT ('0',10X,'Q1COEF EQUATION SYSTEM, ROW ',I2,/,                        QLDR2610
   1' ',10X,'C1COF(',I2,') = ', E14.7,',',E14.7)                                QLDR2615
    DO 2 J = 1,N                                                                QLDR2620
    J2=J+N                                                                      QLDR2625
    WRITE(6,91) I,J,Q1INT(I,J),I,J2,C1INT(I,J2)                                 QLDR2630
 91 FORMAT(' ',15X,2('QINT(',I2,',',I2,') = ',E14.7,',',E14.7,10X))             QLDR2635
  2 CONTINUE                                                                    QLDR2640
```

```
    5 IA = 26                                                                 QLDR2645
      IJOB=0                                                                  QLDR2650
      CALL LEQT1C(Q1INT,N2,IA,Q1COEF,M,IB,IJOB,ZWA,IER)                       QLDR2655
      IF(IER.EQ.0) GOTO 30                                                    QLDR2660
      IF (IER.EQ.129) GO TO 10                                                QLDR2665
      WRITE(6,93)                                                             QLDR2670
   93 FORMAT( '0',10X,'Q1COEF - ITERATIVE IMPROVEMENT FAILED, MATRIX TOO      QLDR2675
     1 ILL-CONDITIONED. USE RESULTS WITH CAUTION.')                           QLDR2680
      GO TO 30                                                                QLDR2685
   10 WRITE (6,95)                                                            QLDR2690
   95 FORMAT('0',10X,'Q1COEF - MATRIX ALGORITHMICALLY SINGULAR.  COEFFIC      QLDR2695
     1IENTS SET TO ZERO.')                                                    QLDR2700
      ZERO = DCMPLX(0.0D0,0.0D0)                                              QLDR2705
      DO 20 I = 1,N                                                           QLDR2710
      I2 = I+N                                                                QLDR2715
      Q1COF(I) = ZERO                                                         QLDR2720
      Q1COF(I2) = ZERO                                                        QLDR2725
   20 CONTINUE                                                                QLDR2730
   30 CONTINUE                                                                QLDR2735
      DO 35 I = 1,N                                                           QLDR2740
      IN = I + N                                                              QLDR2745
      Q1ABCF(I) = Q1COF(IN)                                                   QLDR2750
      Q1RBCF(I) = Q1COF(I)                                                    QLDR2755
   35 CONTINUE                                                                QLDR2760
      IF(IPT.LE.0) RETURN                                                     QLDR2765
      IF (IPT.GE.0) WRITE (6,94)                                              QLDR2770
      DO 40 I = 1,N                                                           QLDR2775
      IM1 = I-1                                                               QLDR2780
      WRITE (6,99) I, IM1, Q1RBCF(I), Q1ABCF(I)                               QLDR2785
   40 CONTINUE                                                                QLDR2790
   99 FORMAT('0',5X,I2,10X,I2,10X,2(E14.7,5X,E14.7,10X))                      QLDR2795
   90 FORMAT( '0',10X,'SUBROUTINE Q1COEF -  COMPLEX POWER SERIES COEFFIC      QLDR2800
     1IENTS')                                                                 QLDR2805
   94 FORMAT( '0',5X,'INDEX',7X,'DEG FOLY',4X,'REFERENCE BLADE TERMS Q1C      QLDR2810
     3OEF(INDEX)',7X,'ADJACENT BLADE TERMS Q1COF(2*INDEX)')                   QLDR2815
      RETURN                                                                  QLDR2820
      END                                                                     QLDR2825
      FUNCTION IQ1IFC(I)                                                      QLDR2830
C     ..................................................................      QLDR2835
C                                                                             QLDR2840
C     IQ1IFC IS A FUNCTION SUBROUTINE DESIGNED TO PRODUCE FACTORIALS          QLDR2845
C     ..................................................................      QLDR2850
      IF(I.EQ.0) GO TO 11                                                     QLDR2855
      IF(I.EQ.1) GO TO 11                                                     QLDR2860
      IQ1IFC=I                                                                QLDR2865
      II=I-1                                                                  QLDR2870
      DO 10 J=1,II                                                            QLDR2875
   10 IQ1IFC=IQ1IFC*(I-J)                                                     QLDR2880
```

```
      GO TO 12                                                                  QLDR2885
   11 IC1IFC=1                                                                  QLDR2890
   12 RETURN                                                                    QLDR2895
      END                                                                       QLDR2900
      SUBROUTINE C1PPOT(DK,DR,DW,RHO,OFFSET,SIGMA,N,X,Q1ABCF,C1RBCF,IPT)        QLDR2905
      IMPLICIT REAL * 8(A-H,O,P,R-Y), COMPLEX * 16(Z,Q)                         QLDR2910
      DIMENSION Q1ABCF(13), Q1RBCF(13), QRR(13), QAA(13), QPHI(13)              QLDR2915
      DIMENSION QDRR(13),QDAA(13),QDPHI(13)                                     QLDR2920
      DIMENSION X(13)                                                           QLDR2925
      IF(IPT.LE.0) RETURN                                                       QLDR2930
      WRITE (6,990) DK,DR,DW,RHO,OFFSET,SIGMA,N,IPT                             QLDR2935
  990 FORMAT('1',10X,'Q1PPOT -CALCULATION OF COMPLEX POTENTIALS FOR ZONE        QLDR2940
     1 ONE',/,'0',10X,'THESE CALCULATIONS ARE BASED ON THE SPANNING FUNC        QLDR2945
     2TION COEFFICIENTS GIVEN DIRECTLY ABOVE',/                                 QLDR2950
     3 ,' ',10X,'DK',13X,'DR',13X,'DK',13X,'RHO',12X,'OFFSET',9X,'SIGMA'        QLDR2955
     4 ,10X,'N',3X,'IPT',/,' ',10X,                                             QLDR2960
     56(E12.5,3X),I2,2X,I2)                                                     QLDR2965
   90 FORMAT ('0 X-STATION NUMBER ',I2,', XSTN = ',F6.4,/,                      QLDR2970
     1'   BL TOTAL POTENTIAL',14X,'REF BL POTENTIAL',8X,                       QLDR2975
     2 'ADJ BL POTENTIAL',8X,'REF BL INT POTENTIAL',5X,                         QLDR2980
     3 'ADJ BL INT POTENTIAL')                                                  QLDR2985
   91 FORMAT('   R  ',2(E14.7,','),8(E11.4,','))                              QLDR2990
   92 FORMAT('   A  ',2(E14.7,','),8(E11.4,','))                              QLDR2995
      ICT = IPT - 1                                                             QLDR3000
      RZERO = 0.0D0                                                             QLDR3005
      DO 10 I = 1,N                                                             QLDR3010
      XSTN = X(I)                                                               QLDR3015
      QRO = Q1PRBP(DK,DR,DW,RZERO,XSTN,IOT)                                     QLDR3020
      QRP = Q1PRBP(DK,DR,DW,RHO,XSTN,IOT)                                       QLDR3025
      QAO = Q1PABP(DK,DR,DW,RZERO,OFFSET,SIGMA,XSTN,IOT)                        QLDR3030
      QAP = Q1PABP(DK,DR,DW,RHO,OFFSET,SIGMA,XSTN,ICT)                          QLDR3035
      QRIPO = Q1PRIP(DK,DR,DW,RZERO,OFFSET,XSTN,C1RBCF,N,IOT)                   QLDR3040
      QRIPP = Q1PRIP(DK,DR,DW,RHO,OFFSET,XSTN,Q1RBCF,N,ICT)                     QLDR3045
      QAIPO = Q1PAIP(DK,DR,DW,RZERO,OFFSET,SIGMA,XSTN,Q1ABCF,N,ICT)             QLDR3050
      QAIPP = Q1PAIP(DK,DR,DW,RHO,OFFSET,SIGMA,XSTN,C1ABCF,N,IOT)               QLDR3055
      QR = QRO + QAP + QRIPO + QAIPP                                            QLDR3060
      QA = QRP + QAO + QRIPP + QAIPO                                            QLDR3065
      QPHI(I)=Q1PHI(DK,DR,DW,RHO,OFFSET,SIGMA,N,Q1ABCF,C1RBCF,XSTN,ICT)         QLDR3070
      WRITE (6,90) I,XSTN                                                       QLDR3075
      WRITE(6,91) QR,QRO,QAP,QRIPO,QAIPP                                        QLDR3080
      WRITE(6,92) QA,QRP,QAO,QRIPP,QAIPO                                        QLDR3085
      QRR(I) = QR                                                               QLDR3090
      QAA(I) = QA                                                               QLDR3095
      QDRO = Q1DRBP(DK,DR,DW,RZERO,XSTN,IOT)                                    QLDR3100
      QDRP = Q1DRBP(DK,DR,DW,RHO,XSTN,ICT)                                      QLDR3105
      QDAO = Q1DABP(DK,DR,DW,RZERO,OFFSET,SIGMA,XSTN,IOT)                       QLDR3110
      QDAP = Q1DABP(DK,DR,DW,RHO,OFFSET,SIGMA,XSTN,ICT)                         QLDR3115
      QDRIPO = Q1DRIP(DK,DR,DW,RZERO,OFFSET,XSTN,C1RBCF,N,IOT)                  QLDR3120
```

```
      QDRIPP = Q1DRIP(DK,DR,DW,RHO,OFFSET,XSTN,C1RBCF,N,IOT)                    QLDR3125
      QDAIPO = Q1DAIP(DK,DR,DW,RZERO,OFFSET,SIGMA,XSTN,Q1ABCF,N,IOT)           QLDR3130
      QDAIPP = Q1DAIP(DK,DR,DW,RHO,OFFSET,SIGMA,XSTN,Q1ABCF,N,IOT)             QLDR3135
      QDR = QDRO + QDAP + QDRIPO + QDAIPP                                      QLDR3140
      QDA = QDRP + QDAO + QDRIPP + QDAIPO                                      QLDR3145
      QDPHI(I)=Q1DPHI(DK,DR,DW,RHO,OFFSET,SIGMA,N,C1ABCF,C1RBCF,               QLDR3150
     1     XSTN,IOT)                                                           QLDR3155
      WRITE (6,95) I,XSTN                                                      QLDR3160
      WRITE(6,91) QDR,QDRO,QDAP,QDRIPO,QDAIPP                                  QLDR3165
      WRITE(6,92) QDA,QDRP,QDAO,QDRIPP,QDAIPO                                  QLDR3170
      QDRR(I) = QDR                                                            QLDR3175
      QDAA(I) = QDA                                                            QLDR3180
   95 FORMAT ('0 X-STATION NUMBER ',I2,', XSTN = ',F6.4,/,                     QLDR3185
     1'  BL TOTAL D(POT)/DX',14X,'REF BL D(POT)/DX',8X,                       QLDR3190
     2 'ADJ BL D(POT)/DX',8X,'REF BL INT D(POT)/DX',5X,                       QLDR3195
     3 'ADJ BL INT D(POT)/DX')                                                QLDR3200
   10 CONTINUE                                                                QLDR3205
      WRITE(6,994)                                                           QLDR3210
  994 FORMAT ('1',10X,'SUMMARY LISTING',/,                                   QLDR3215
     1'0',10X,'XSTN',7X,'SINGLE BLADE TOTAL POTENTIAL',6X,                   QLDR3220
     1  'REF BLADE POTENTIAL',15X,'ADJ BLADE POTENTIAL')                      QLDR3225
      DO 20 I = 1,N                                                          QLDR3230
      XSTN = X(I)                                                            QLDR3235
      WRITE(6,94) XSTN,QPHI(I),QRR(I),QAA(I)                                 QLDR3240
   94 FORMAT(' ',10X,F6.4,3(5X,E14.7,',',E14.7))                             QLDR3245
   20 CONTINUE                                                               QLDR3250
      WRITE(6,996)                                                           QLDR3255
  996 FORMAT('0',10X,'XSTN',7X,'SINGLE BLADE TOTAL D(POT)/DX',6X,            QLDR3260
     1  'REF BLADE D(POT)/DX',15X,'ADJ BLADE D(POT)/DX')                      QLDR3265
      DO 50 I = 1,N                                                          QLDR3270
      XSTN = X(I)                                                            QLDR3275
      WRITE(6,94) XSTN,QDPHI(I),QDRR(I),QDAA(I)                              QLDR3280
   50 CONTINUE                                                               QLDR3285
      RETURN                                                                 QLDR3290
      END                                                                    QLDR3295
      SUBROUTINE Q1DCOF(DK,DR,DW,RHO,OFFSET,SIGMA,N,Q1ABCF,Q1RBCF,IPT)       QLDR3300
      IMPLICIT REAL * 8 (A-H,O,P,R-Y), COMPLEX * 16 (Q,Z)                    QLDR3305
      DIMENSION Q1ABCF(13), Q1RBCF(13)                                       QLDR3310
      IF(IPT.GT.0)WRITE (6,990) DK,DR,DW,RHO,OFFSET,SIGMA,N,IPT              QLDR3315
  990 FORMAT('1',10X,'Q1DCOF - CALCULATION OF COMPLEX DIMENSIONLESS AERO     QLDR3320
     1DYNAMIC COEFFICIENTS',/,                                               QLDR3325
     3  '0',10X,'DK',13X,'DR',13X,'DW',13X,'RHO',12X,'OFFSET',9X,'SIGMA'     QLDR3330
     4  ,10X,'N',3X,'IPT',/,' ',10X,                                        QLDR3335
     56(E12.5,3X),I2,2X,I3)                                                  QLDR3340
      ICT = IPT - 3                                                          QLDR3345
  999 QDCL = Q1DCL(DK,DR,DW,RHO,OFFSET,SIGMA,N,Q1ABCF,C1RBCF,IOT)            QLDR3350
      QDCM = Q1DCM(DK,DR,DW,RHO,OFFSET,SIGMA,N,Q1ABCF,Q1RBCF,IOT)            QLDR3355
      GAMMA = 1.400                                                          QLDR3360
```

```
      TAU = (2.0D0*DSQRT((GAMMA + 1.0D0)*DW))/DR                        QLDR3365
      WRITE (6,90) DK,TAU,DW,N,SIGMA,CDCL,CDCM                          QLDR3370
C     FORMAT('0',5X,'DK = ',F6.3,', TAU = ',F7.4,', DW = ',F6.3,', N = 'QLDR3375
     1      ,I2,', SIGMA = ',F6.3,', CL = ',F9.4,',',F9.4,', CM = ',    QLDR3380
     2      F9.4,',',F9.4)                                              QLDR3385
      RETURN                                                            QLDR3390
      END                                                               QLDR3395
      COMPLEX FUNCTION Q1DCL*16(DK,DR,DW,RHO,OFFSET,SIGMA,N,Q1ABCF,Q1RBCQLDR3400
     1F,IPT)                                                            QLDR3405
      IMPLICIT REAL*8(A - E,G,H,O,P,R - Y), COMPLEX*16(F,Q,Z)           QLDR3410
      DIMENSION Q1ABCF(13), Q1RBCF(13)                                  QLDR3415
      DIMENSION FV(5),F1T(60),F2T(60),F3T(60),QEST1(60),CPSUM(60)       QLDR3420
      DIMENSION DAT(60),   AREST1(60),EPST(60)                          QLDR3425
      DIMENSION LCRR(60)                                                QLDR3430
      F(X) = (Q1DPHI(DK,DR,DW,RHO,OFFSET,SIGMA,N,Q1ABCF,Q1RBCF,X,ICT)   QLDR3435
     1    -QALPHA*Q1PHI(DK,DR,DW,RHO,OFFSET,SIGMA,N,Q1ABCF,Q1RBCF,X,ICT))QLDR3440
      IF (IPT.GT.0) WRITE (6,990) DK,DR,DW,RHO,OFFSET,SIGMA,N,IPT       QLDR3445
0     FORMAT('0',10X,'Q1DCL ENTERED WITH:',/,                           QLDR3450
     3 ' ',10X,'DK',13X,'DR',13X,'DW',13X,'RHO',12X,'OFFSET',9X,'SIGMA' QLDR3455
     4 ,10X,'N',3X,'IPT',/,' ',10X,                                     QLDR3460
     5 6(E12.5,3X),I2,2X,I3)                                            QLDR3465
      A = - 1.0D0                                                       QLDR3470
      B = 1.0D0                                                         QLDR3475
      ICT = IPT - 1                                                     QLDR3480
      GAMMA = 1.4D0                                                     QLDR3485
      CLAMDA = DK/((GAMMA + 1.0D0) * DW)                                QLDR3490
      QALPHA = DCMPLX(0.0D0,CLAMDA-DK)                                  QLDR3495
      U   =  9.0E - 13                                                  QLDR3500
      ACC = 1.0D-5                                                      QLDR3505
      EPSURU = 4.0*U                                                    QLDR3510
      IFLAG = 1                                                         QLDR3515
      EPS = ACC                                                         QLDR3520
      QERROR = DCMPLX(0.0D0,0.0D0)                                      QLDR3525
      LVL = 1                                                           QLDR3530
      LCRR(LVL) = 1                                                     QLDR3535
      QPSUM(LVL) = 0.0                                                  QLDR3540
      ALPHA = A                                                         QLDR3545
      DA = B - A                                                        QLDR3550
      AREA = 0.0                                                        QLDR3555
      AREST = 0.0                                                       QLDR3560
      FV(1) = F(ALPHA)                                                  QLDR3565
      FV(3) = F(ALPHA + 0.5*DA)                                         QLDR3570
      FV(5) = F(ALPHA + DA)                                             QLDR3575
      KCNT = 3                                                          QLDR3580
      WT = DA/6.0                                                       QLDR3585
      QEST = WT*(FV(1) + 4.0*FV(3) + FV(5))                             QLDR3590
      DX = 0.5*DA                                                       QLDR3595
      FV(2) = F(ALPHA + 0.5*DX)                                         QLDR3600
```

```
      FV(4) = F(ALPHA + 1.5*DX)                                                 QLDR3605
      KOUNT = KOUNT + 2                                                         QLDR3610
      WT = DX/6.0                                                               QLDR3615
      QESTL = WT*(FV(1) + 4.0*FV(2) + FV(3))                                    QLDR3620
      QESTR = WT*(FV(3) + 4.0*FV(4) + FV(5))                                    QLDR3625
      QSUM = QESTL + QESTR                                                      QLDR3630
      ARESTL = WT*(CDABS(FV(1)) + CDABS(4.0*FV(2)) + CDABS(FV(3)))              QLDR3635
      ARESTR = WT*(CDABS(FV(3)) + CDABS(4.0*FV(4)) + CDABS(FV(5)))              QLDR3640
      AREA = AREA + ((ARESTL + ARESTR) - AREST)                                 QLDR3645
      QDIFF = QEST - QSUM                                                       QLDR3650
      IF(CDABS(QDIFF).LE.EPS*DABS(AREA))GO TO 2                                 QLDR3655
      IF(DABS(DX).LE.EPSILON*DABS(ALPHA))GO TO 5                                QLDR3660
      IF(LVL.GE.60)GO TO 5                                                      QLDR3665
      IF(KOUNT.GE.4000)GO TO 6                                                  QLDR3670
      LVL = LVL + 1                                                             QLDR3675
      LCRR(LVL) = 0                                                             QLDR3680
      F1T(LVL) = FV(3)                                                          QLDR3685
      F2T(LVL) = FV(4)                                                          QLDR3690
      F3T(LVL) = FV(5)                                                          QLDR3695
      DA = DX                                                                   QLDR3700
      DAT(LVL) = DX                                                             QLDR3705
      AREST = ARESTL                                                            QLDR3710
      ARESTT(LVL) = ARESTR                                                      QLDR3715
      QEST = QESTL                                                              QLDR3720
      QESTT(LVL) = QESTR                                                        QLDR3725
      EPS = EPS/1.4                                                             QLDR3730
      EPST(LVL) = EPS                                                           QLDR3735
      FV(5) = FV(3)                                                             QLDR3740
      FV(3) = FV(2)                                                             QLDR3745
      GO TO 1                                                                   QLDR3750
2     QERROR = QERROR + QDIFF/15.0                                              QLDR3755
3     IF(LCRR(LVL).EQ.0)GO TO 4                                                 QLDR3760
      QSUM = QPSUM(LVL) + QSUM                                                  QLDR3765
      LVL = LVL - 1                                                             QLDR3770
      IF(LVL.GT.1)GO TO 3                                                       QLDR3775
      Q1DCL = QSUM * 2.0D0                                                      QLDR3780
      IF(IPT.GT.0) GO TO 11                                                     QLDR3785
      IF(IFLAG.EQ.1) RETURN                                                     QLDR3790
      WRITE(6,990) DK,DR,DW,RHO,OFFSET,SIGMA,N,IPT                              QLDR3795
   11 WRITE(6,995) Q1DCL,IFLAG,QERROR                                           QLDR3800
  995 FORMAT(' ',15X,'RESULTS: Q1DCL = ',E14.7,',',E14.7,/,                     QLDR3805
     1' ',15X,'IFLAG',2X,'QERROR',/,                                            QLDR3810
     2' ',15X,I3,2X,E14.7,',',E14.7)                                            QLDR3815
      RETURN                                                                    QLDR3820
4     QPSUM(LVL) = QSUM                                                         QLDR3825
      LCRR(LVL) = 1                                                             QLDR3830
      ALPHA = ALPHA + DA                                                        QLDR3835
      DA = DAT(LVL)                                                             QLDR3840
```

```
      FV(1) = F1T(LVL)                                                  QLDR3845
      FV(3) = F2T(LVL)                                                  QLDR3850
      FV(5) = F3T(LVL)                                                  QLDR3855
      AREST = ARESTT(LVL)                                               QLDR3860
      QEST = QESTT(LVL)                                                 QLDR3865
      EPS = EPST(LVL)                                                   QLDR3870
      GO TO 1                                                           QLDR3875
    5 IFLAG = 2                                                         QLDR3880
      GO TO 2                                                           QLDR3885
    6 IFLAG = 3                                                         QLDR3890
      GO TO 2                                                           QLDR3895
      END                                                               QLDR3900
      COMPLEX FUNCTION C1DCM*16(DK,DR,DW,RHO,OFFSET,SIGMA,N,Q1ABCF,Q1RBCQLDR3905
     1F,IPT)                                                            QLDR3910
      IMPLICIT REAL*8(A - E,G,H,O,P,R - Y), COMPLEX*16(F,C,Z)           QLDR3915
      DIMENSION Q1ABCF(13), Q1RBCF(13)                                  QLDR3920
      DIMENSION FV(5),F1T(60),F2T(60),F3T(60),QESTT(60),CFSUM(60)       QLDR3925
      DIMENSION DAT(60),  ARESTT(60),EPST(60)                           QLDR3930
      DIMENSION LCRR(60)                                                QLDR3935
      F(X) = X*(Q1DPHI(DK,DR,DW,RHO,OFFSET,SIGMA,N,Q1ABCF,Q1RBCF,X,IOT) QLDR3940
     1    -QALPHA*C1PHI(DK,DR,DW,RHO,OFFSET,SIGMA,N,Q1ABCF,Q1RBCF,X,IOT))QLDR3945
      IF (IPT.GT.0) WRITE (6,990) DK,DR,DW,RHO,OFFSET,SIGMA,N,IPT       QLDR3950
  990 FORMAT('0',10X,'Q1DCM ENTERED WITH:',/,                           QLDR3955
     3 ' ',10X,'DK',13X,'DR',13X,'DK',13X,'RHO',12X,'OFFSET',9X,'SIGMA' QLDR3960
     4 ,10X,'N',3X,'IPT',/,' ',10X,                                     QLDR3965
     56(E12.5,3X),I2,2X,I3)                                             QLDR3970
      A = - 1.0D0                                                       QLDR3975
      B = 1.0D0                                                         QLDR3980
      IOT = IPT - 1                                                     QLDR3985
      GAMMA = 1.4D0                                                     QLDR3990
      CLAMDA = DK/((GAMMA + 1.0D0) * DW)                                QLDR3995
      QALPHA = DCMPLX(0.0D0,CLAMDA-DK)                                  QLDR4000
      U  =  9.0E - 13                                                   QLDR4005
      ACC = 1.0D-5                                                      QLDR4010
      EFOURU = 4.0*U                                                    QLDR4015
      IFLAG = 1                                                         QLDR4020
      EPS = ACC                                                         QLDR4025
      CERROR = DCMPLX(0.0D0,0.0D0)                                      QLDR4030
      LVL = 1                                                           QLDR4035
      LCRR(LVL) = 1                                                     QLDR4040
      CFSUM(LVL) = 0.0                                                  QLDR4045
      ALPHA = A                                                         QLDR4050
      DA = B - A                                                        QLDR4055
      AREA = C.0                                                        QLDR4060
      AREST = 0.0                                                       QLDR4065
      FV(1) = F(ALPHA)                                                  QLDR4070
      FV(3) = F(ALPHA + 0.5*DA)                                         QLDR4075
      FV(5) = F(ALPHA + DA)                                             QLDR4080
```

```
      KCUNT = 3                                                          QLDR4085
      WT = DA/6.0                                                        QLDR4090
      QEST = WT*(FV(1) + 4.0*FV(3) + FV(5))                              QLDR4095
    1 DX = 0.5*DA                                                        QLDR4100
      FV(2) = F(ALPHA + 0.5*DX)                                          QLDR4105
      FV(4) = F(ALPHA + 1.5*DX)                                          QLDR4110
      KOUNT = KOUNT + 2                                                  QLDR4115
      WT = DX/6.0                                                        QLDR4120
      QESTL = WT*(FV(1) + 4.0*FV(2) + FV(3))                             QLDR4125
      QESTR = WT*(FV(3) + 4.0*FV(4) + FV(5))                             QLDR4130
      QSUM = QESTL + QESTR                                               QLDR4135
      ARESTL = WT*(CDABS(FV(1)) + CDABS(4.0*FV(2)) + CDABS(FV(3)))       QLDR4140
      ARESTR = WT*(CDABS(FV(3)) + CDABS(4.0*FV(4)) + CDABS(FV(5)))       QLDR4145
      AREA = AREA + ((ARESTL + ARESTR) - AREST)                          QLDR4150
      QDIFF = QEST - QSUM                                                QLDR4155
      IF(CDABS(QDIFF).LE.EPS*DABS(AREA))GO TO 2                          QLDR4160
      IF(DABS(DX).LE.EFCURU*DABS(ALPHA))GO TO 5                          QLDR4165
      IF(LVL.GE.60)GO TO 5                                               QLDR4170
      IF(KOUNT.GE.4000)GO TO 6                                           QLDR4175
      LVL = LVL + 1                                                      QLDR4180
      LORR(LVL) = 0                                                      QLDR4185
      F1T(LVL) = FV(3)                                                   QLDR4190
      F2T(LVL) = FV(4)                                                   QLDR4195
      F3T(LVL) = FV(5)                                                   QLDR4200
      DA = DX                                                            QLDR4205
      DAT(LVL) = DX                                                      QLDR4210
      AREST = ARESTL                                                     QLDR4215
      ARESTT(LVL) = ARESTR                                               QLDR4220
      QEST = QESTL                                                       QLDR4225
      QESTT(LVL) = QESTR                                                 QLDR4230
      EPS = EPS/1.4                                                      QLDR4235
      EPST(LVL) = EPS                                                    QLDR4240
      FV(5) = FV(3)                                                      QLDR4245
      FV(3) = FV(2)                                                      QLDR4250
      GO TO 1                                                            QLDR4255
    2 QERRCR = QERROR + QDIFF/15.0                                       QLDR4260
    3 IF(LORR(LVL).EQ.0)GO TO 4                                          QLDR4265
      QSUM = QPSUM(LVL) + QSUM                                           QLDR4270
      LVL = LVL - 1                                                      QLDR4275
      IF(LVL.GT.1)GO TO 3                                                QLDR4280
      Q1DCM = QSUM * 2.0D0                                               QLDR4285
      IF(IPT.GT.0) GO TO 11                                              QLDR4290
      IF(IFLAG.EQ.1) RETURN                                              QLDR4295
      WRITE(6,990) DK,DR,DW,RHO,OFFSET,SIGMA,N,IPT                       QLDR4300
   11 WRITE(6,995) Q1DCM,IFLAG,QERRCR                                    QLDR4305
  995 FORMAT(' ',15X,'RESULTS: Q1DCM = ',E14.7,',',E14.7,/,              QLDR4310
     1' ',15X,'IFLAG',2X,'QERROR',/,                                     QLDR4315
     2' ',15X,I3,2X,E14.7,',',E14.7)                                     QLDR4320
```

```
      RETURN                                                            QLDR4325
4     QPSUM(LVL) = QSUM                                                 QLDR4330
      LCRR(LVL) = 1                                                     QLDR4335
      ALPHA = ALPHA + DA                                                QLDR4340
      DA = DAT(LVL)                                                     QLDR4345
      FV(1) = F1T(LVL)                                                  QLDR4350
      FV(3) = F2T(LVL)                                                  QLDR4355
      FV(5) = F3T(LVL)                                                  QLDR4360
      AREST = ARESTT(LVL)                                               QLDR4365
      QEST = QESTT(LVL)                                                 QLDR4370
      EPS = EPST(LVL)                                                   QLDR4375
      GO TO 1                                                           QLDR4380
5     IFLAG = 2                                                         QLDR4385
      GO TO 2                                                           QLDR4390
6     IFLAG = 3                                                         QLDR4395
      GO TO 2                                                           QLDR4400
      END                                                               QLDR4405
      COMPLEX FUNCTION C1PHI*16 (DK,DR,DW,RHO,OFFSET,SIGMA,             QLDR4410
     1    N,Q1ABCF,Q1RBCF,XSTN,IPT)                                     QLDR4415
      IMPLICIT REAL*8(A-H,O,P,R-Y), COMPLEX*16(Q,Z)                     QLDR4420
      DIMENSION Q1ABCF(13), Q1RBCF(13)                                  QLDR4425
      IF (IPT.GT.0) WRITE(6,990) DK,DR,DW,RHO,OFFSET,SIGMA,XSTN,IPT     QLDR4430
990   FORMAT('0',10X,'C1PHI ENTERED WITH:',/,                           QLDR4435
     1  ' ',10X,'DK',11X,'DR',11X,'DW',11X,'RHO',10X,'OFFSET',7X,'SIGMA'QLDR4440
     2  ,8X,'XSTN',9X,'IPT'/,' ',10X,7(E12.5,','),I3)                  QLDR4445
      RZERO = 0.0D0                                                     QLDR4450
      ICT = IPT - 1                                                     QLDR4455
      C1PHI = Q1PRBP(DK,DR,DW,RZERO,XSTN,IOT)                           QLDR4460
     1  + Q1PRIP(DK,DR,DW,RZERO,OFFSET,XSTN,Q1RBCF,N,IOT)               QLDR4465
     2  + C1PABP(DK,DR,DW,DR,OFFSET,SIGMA,XSTN,IOT)                     QLDR4470
     3  + Q1PAIP(DK,DR,DW,DR,OFFSET,SIGMA,XSTN,Q1ABCF,N,ICT)            QLDR4475
      IF(IPT.LE.0) RETURN                                               QLDR4480
      WRITE (6,995) Q1PHI                                               QLDR4485
995   FORMAT('0',10X,'C1PHI = ',E14.7,',',E14.7)                        QLDR4490
      RETURN                                                            QLDR4495
      END                                                               QLDR4500
      COMPLEX FUNCTION C1PRBP*16(DK,DR,DW,RHO,XSTN,IPT)                 QLDR4505
      IMPLICIT REAL * 8 (A-H,O,P,R-Y), COMPLEX * 16 (Q,Z)               QLDR4510
      DIMENSION QINP(2)                                                 QLDR4515
      IF (IPT.GT.0) WRITE(6,990) DK,DR,DW,RHO,XSTN,IPT                  QLDR4520
990   FORMAT('0',10X,'C1PRBP ENTERED WITH:',/,                          QLDR4525
     1  ' ',10X,'DK',11X,'DR',11X,'DW',11X,'RHO',10X,'XSTN',9X,'IPT'    QLDR4530
     2  ,/,' ',10X,5(E12.5,','),I3)                                    QLDR4535
      IF(XSTN.LE.RHO-1.0D0) GOTO 20                                     QLDR4540
      ICT = IPT - 1                                                     QLDR4545
      CDK = DCMPLX (0.0D0,DK)                                           QLDR4550
      CALL C1PWXM(CDK,DR,DW,RHO,XSTN,CINF,IOT)                          QLDR4555
      C1PRBP = -QDK*QINF(2) - QINP(1)                                   QLDR4560
```

```
      IF(IPT.LE.0) RETURN                                               QLDR4565
      GOTO 60                                                           QLDR4570
   20 Q1PRBP = DCMPLX(C.CDO,0.0DO)                                      QLDR4575
      IF(IPT.LE.0) RETURN                                               QLDR4580
   60 WRITE (6,995) Q1PRBP                                              QLDR4585
  995 FORMAT('0',10X,'Q1PRBP = ',E14.7,',',E14.7)                       QLDR4590
      RETURN                                                            QLDR4595
      END                                                               QLDR4600
      COMPLEX FUNCTION Q1PABP*16(DK,DR,DW,RHO,OFFSET,SIGMA,XSTN,IPT)    QLDR4605
      IMPLICIT REAL * 8 (A-H,C,P,R-Y), COMPLEX * 16 (Q,Z)               QLDR4610
      DIMENSION QINP(2)                                                 QLDR4615
      IF(IPT.GT.0) WRITE(6,990) DK,DR,DW,RHO,OFFSET,SIGMA,XSTN,IPT      QLDR4620
  990 FORMAT('C',10X,'Q1PABP ENTERED WITH:',/,                          QLDR4625
     1   ' ',10X,'DK',11X,'DR',11X,'DW',11X,'RHO',10X,'OFFSET',7X,     QLDR4630
     2   'SIGMA',8X,'XSTN',9X,'IPT'/,' ',10X,7(E12.5,','),I3)          QLDR4635
      XASTN = XSTN - OFFSET                                             QLDR4640
      IF(XASTN.LE.RHO-1.0DO) GOTO 20                                    QLDR4645
      ICT = IPT - 1                                                     QLDR4650
      CDK = DCMPLX (0.0DO,DK)                                           QLDR4655
      QCONST = CDEXP( DCMPLX(0.0DO,SIGMA))                              QLDR4660
      CALL Q1PWXM(DK,DR,DW,RHO,XASTN,QINP,IOT)                          QLDR4665
      Q1PABP = (QDK*QINP(2) + QINP(1)) * QCONST                         QLDR4670
      IF(IPT.LE.0) RETURN                                               QLDR4675
      GO TO 60                                                          QLDR4680
   20 Q1PABP = DCMPLX(C.CDO,0.0DO)                                      QLDR4685
      IF (IPT.LE.0) RETURN                                              QLDR4690
   60 WRITE (6,995) Q1PABP                                              QLDR4695
  995 FORMAT('0',10X,'Q1PABP = ',E14.7,',',E14.7)                       QLDR4700
      RETURN                                                            QLDR4705
      END                                                               QLDR4710
      SUBROUTINE Q1PWXM(DK,DR,DW,RHO,XSTN,QINP,IPT)                     QLDR4715
C                                                                       QLDR4720
CC    N IS THE MAXIMUM DEGREE OF THE TERM U IN THE INTEGRALS            QLDR4725
C                                                                       QLDR4730
      IMPLICIT REAL * 8 (A-H,C,P,R-Y), COMPLEX * 16 (Z,Q)               QLDR4735
      DIMENSION QINP(2)                                                 QLDR4740
      IF(IPT.GT.0) WRITE(6,990) DK,DR,RHO,XSTN,IPT                      QLDR4745
  990 FORMAT('C',10X,'Q1PWXM ENTERED WITH ARGUMENTS:',/,                QLDR4750
     1' ',10X,'DK',16X,'DR',16X,'RHO',15X,'XSTN',14X,'IPT',/,           QLDR4755
     2' ',10X,4(E13.6,5X),I2,5X,I3)                                     QLDR4760
      IOT = IPT - 1                                                     QLDR4765
      B = XSTN - RHO - 1.0D-10                                          QLDR4770
      A = -1.0DO                                                        QLDR4775
      CALL Q1PZXM(DK,DR,DW,RHO,XSTN,A,B,1,QANS,IOT)                     QLDR4780
      QINP(1) = QANS                                                    QLDR4785
      CALL Q1PZXM(DK,DR,DW,RHO,XSTN,A,B,2,QANS,IOT)                     QLDR4790
      QINP(2) = QANS                                                    QLDR4795
      IF(IPT.LE.0) RETURN                                               QLDR4800
```

```
      WRITE (6,995)                                                     QLDR4805
  995 FORMAT(' ',10X,'Q1PWXM RESULTS:',/,                               QLDR4810
     1' ',11X,'J',6X,'QINP(I)')                                         QLDR4815
      DO 1001 I = 1,2                                                   QLDR4820
  996 FORMAT(' ',10X,I3,5X,E12.6,',',E12.6)                             QLDR4825
 1001 WRITE (6,996) I, QINP(I)                                          QLDR4830
      RETURN                                                            QLDR4835
      END                                                               QLDR4840
      SUBROUTINE Q1PZXM (DK,DR,DW,RHO,XSTN,A,B,J,QANS,IPT)              QLDR4845
      IMPLICIT REAL*8(A - E,G,H,M,O,P,R - Y), COMPLEX*16(F ,Q,Z)        QLDR4850
      DIMENSION FV(5),LCRR(60),F1T(60),F2T(60),F3T(60),DAT(60),         QLDR4855
     1  ARESTT(60),QESTT(60),EPST(60),CPSUM(60)                         QLDR4860
      F(X) = (X**J1)*CDEXP(QEXP*(X))                                    QLDR4865
     1     * MMESJ0((OMEGA * DSQRT( (XSTN-X)*(XSTN-X) - YY)), IER)      QLDR4870
      GAMMA = 1.4D0                                                     QLDR4875
      YY = RHO * RHO                                                    QLDR4880
      DM2 = (GAMMA + 1.0D0) * DW                                        QLDR4885
      OMEGA = DSQRT (DK*DK*(1.0D0-DM2)/(DM2*DM2))                       QLDR4890
      CLAMDA = DK/DM2                                                   QLDR4895
      DM = DSQRT(DM2)                                                   QLDR4900
      C2EXP = CDEXP (DCMPLX (0.0D0, -CLAMDA*XSTN))                      QLDR4905
      QEXP = DCMPLX(0.0D0,CLAMDA)                                       QLDR4910
      J1 = J - 1                                                        QLDR4915
      U  =  9.0D-13                                                     QLDR4920
      ACC = 1.0D-6                                                      QLDR4925
      IF (IPT.GT.0) WRITE(6,990)DK,DR,DW,RHO,XSTN,A,B,J,IPT             QLDR4930
  990 FORMAT('0',15X,'Q1PZXM ENTERED WITH ARGUMENTS:',/,                QLDR4935
     1' ',15X,'DK',13X,'DR',13X,'DW    ',10X,'RHO',12X,'XSTN',11X,     QLDR4940
     2'A',14X,'B',14X,'J',2X,'IPT',/,                                   QLDR4945
     3' ',15X,7(E14.7,','),I2,2X,I3)                                    QLDR4950
      EFCURU = 4.0*U                                                    QLDR4955
      IFLAG = 1                                                         QLDR4960
      EPS = ACC                                                         QLDR4965
      CERROR = DCMPLX(0.0D0,0.0D0)                                      QLDR4970
      LVL = 1                                                           QLDR4975
      LCRR(LVL) = 1                                                     QLDR4980
      CPSUM(LVL) = 0.0                                                  QLDR4985
      ALPHA = A                                                         QLDR4990
      DA = B - A                                                        QLDR4995
      AREA = 0.0                                                        QLDR5000
      ARES = 0.0                                                        QLDR5005
      FV(1) = F(ALPHA)                                                  QLDR5010
      FV(3) = F(ALPHA + 0.5*DA)                                         QLDR5015
      FV(5) = F(ALPHA + DA)                                             QLDR5020
      KOUNT = 3                                                         QLDR5025
      WT = DA/6.0                                                       QLDR5030
      EST = WT*(FV(1) + 4.0*FV(3) + FV(5))                              QLDR5035
     1  DX = 0.5*DA                                                     QLDR5040
```

```
      FV(2) = F(ALPHA + 0.5*DX)                                                 QLDR5C45
      FV(4) = F(ALPHA + 1.5*DX)                                                 QLDR5050
      KCUNT = KCUNT + 2                                                         QLDR5C55
      WT = DX/6.0                                                               QLDR5060
      QESTL = WT*(FV(1) + 4.0*FV(2) + FV(3))                                    QLDR5C65
      QESTR = WT*(FV(3) + 4.0*FV(4) + FV(5))                                    QLDR507C
      QSUM = QESTL + QESTR                                                      QLDR5075
      ARESTL = WT*(CDABS(FV(1)) + CDABS(4.0*FV(2)) + CDABS(FV(3)))              QLDR5080
      ARESTR = WT*(CDABS(FV(3)) + CDABS(4.0*FV(4)) + CDABS(FV(5)))              QLDR5C85
      AREA = AREA + ((ARESTL + ARESTR) - AREST)                                 QLDR5090
      CDIFF = QEST - QSUM                                                       QLDR5095
      IF(CDABS(QDIFF).LE.EPS*DABS(AREA))GO TO 2                                 QLDR51CC
      IF(DABS(DX).LE.EFOURU*DABS(ALPHA))GO TO 5                                 QLDR5105
      IF(LVL.GE.60)GO TO 5                                                      QLDR511C
      IF(KOUNT.GE.2000)GC TC 6                                                  QLDR5115
      LVL = LVL + 1                                                             QLDR5120
      LCRR(LVL) = 0                                                             QLDR5125
      F1T(LVL) = FV(3)                                                          QLDR5130
      F2T(LVL) = FV(4)                                                          QLDR5135
      F3T(LVL) = FV(5)                                                          QLDR5140
      DA = DX                                                                   QLDR5145
      DAT(LVL) = DX                                                             QLDR5150
      AREST = ARESTL                                                            QLDR5155
      ARESTT(LVL) = ARESTR                                                      QLDR5160
      QEST = QESTL                                                              QLDR5165
      QESTT(LVL) = QESTR                                                        QLDR5170
      EPS = EPS/1.4                                                             QLDR5175
      EPST(LVL) = EPS                                                           QLDR518C
      FV(5) = FV(3)                                                             QLDR5185
      FV(3) = FV(2)                                                             QLDR519C
      GC TO 1                                                                   QLDR5195
2     QERROR = QERROR + CDIFF/15.0                                              QLDR520C
3     IF(LORR(LVL).EQ.0)GC TC 4                                                 QLDR5205
      QSUM = CPSUM(LVL) + QSUM                                                  QLDR5210
      LVL = LVL - 1                                                             QLDR5215
      IF(LVL.GT.1)GO TO 3                                                       QLDR5220
      QANS = QSUM * Q2EXP / DM                                                  QLDR5225
      IF(IPT.GT.0) GO TO 11                                                     QLDR5230
      IF(IFLAG.EC.1) RETURN                                                     QLDR5235
      WRITE(6,990) DK,DR,DW,RHO,XSIN,A,E,J,IPT                                  QLDR5240
 11   WRITE(6,995) QANS,IFLAG,IER,QERR(R                                        QLDR5245
995   FORMAT(' ',15X,'RESULTS: QANS = ',E14.7,',',E14.7,/,                      QLDR5250
     1' ',15X,'IFLAG',2X,'IER',5X,'QERR(R',/,                                   QLDR5255
     2' ',15X,I3,2X,I3,5X,E14.7,',',E14.7)                                      QLDR526C
      RETURN                                                                    QLDR5265
4     CPSUM(LVL) = QSUM                                                         QLDR5270
      LCRR(LVL) = 1                                                             QLDR5275
      ALPHA = ALPHA + DA                                                        QLDR5280
```

```
      DA = DAT(LVL)                                                     QLDR5285
      FV(1) = F1T(LVL)                                                  QLDR5290
      FV(3) = F2T(LVL)                                                  QLDR5295
      FV(5) = F3T(LVL)                                                  QLDR5300
      AREST = ARESTT(LVL)                                               QLDR5305
      QEST = QESTT(LVL)                                                 QLDR5310
      EPS = EPST(LVL)                                                   QLDR5315
      GO TO 1                                                           QLDR5320
    5 IFLAG = 2                                                         QLDR5325
      GO TO 2                                                           QLDR5330
    6 IFLAG = 3                                                         QLDR5335
      GO TO 2                                                           QLDR5340
      END                                                               QLDR5345
      COMPLEX FUNCTION C1PAIP*16(DK,DR,DW,RHO,OFST,SGMA,XSTN,C1CF,N,IPT)QLDR5350
      IMPLICIT REAL*8 (A-H,O,P,R-Y), COMPLEX * 16 (Q,Z)                 QLDR5355
      DIMENSION Q1CF(13)                                                QLDR5360
      IF (IPT.GT.0) WRITE(6,990) DK,DR,DW,RHO,OFST,SGMA,XSTN,IPT        QLDR5365
  990 FORMAT('0',10X,'Q1PAIP ENTERED WITH:',/,                          QLDR5370
     1   ' ',10X,'DK',11X,'DR',11X,'DW',11X,'RHO',10X,'OFST',9X,'SGMA',9XQLDR5375
     2  ,'XSTN',9X,'IPT'/,' ',10X,7(E12.5,','),I3)                     QLDR5380
      XASTN = XSTN - OFST                                               QLDR5385
      IF(XASTN.LE.DR+RHO-1.0D0) GO TO 20                                QLDR5390
      A = DR-1.0D0                                                      QLDR5395
      B = XASTN - RHO - 1.0D-8                                          QLDR5400
      IOT = IPT - 1                                                     QLDR5405
      CCONST = CDEXP( DCMPLX(0.0D0,SGMA))                               QLDR5410
      CALL Q1PZSM(DK,DR,DW,RHO,XASTN,A,B,N,QANS,Q1CF,IOT)               QLDR5415
      C1PAIP = QANS                                                     QLDR5420
      IF(IPT.LE.0) RETURN                                               QLDR5425
      GOTO 60                                                           QLDR5430
   20 C1PAIP = DCMPLX(0.0D0,0.0D0)                                      QLDR5435
      IF(IPT.LE.0)RETURN                                                QLDR5440
   60 WRITE (6,995) Q1PAIP                                              QLDR5445
  995 FORMAT('0',10X,'C1PAIP = ',E14.7,',',E14.7)                       QLDR5450
      RETURN                                                            QLDR5455
      END                                                               QLDR5460
      COMPLEX FUNCTION C1PRIP*16(DK,DR,DW,RHO,OFFSET,XSTN,C1CF,N,IPT)   QLDR5465
      IMPLICIT REAL*8 (A-H,O,P,R-Y), COMPLEX * 16 (Q,Z)                 QLDR5470
      DIMENSION C1CF(13)                                                QLDR5475
      IF (IPT.GT.0) WRITE(6,990) DK,DR,DW,RHO,XSTN,IPT                  QLDR5480
  990 FORMAT('0',10X,'Q1PRIP ENTERED WITH:',/,                          QLDR5485
     1   ' ',10X,'DK',11X,'DR',11X,'DW',11X,'RHO',10X,                  QLDR5490
     2  'XSTN',9X,'IPT'/,' ',10X,5(E12.5,','),I3)                       QLDR5495
      IF(XSTN.LE.DR+RHO+OFFSET-1.0D0) GOTO 20                           QLDR5500
      A = DR-1.0D0                                                      QLDR5505
      B = XSTN - RHO - 1.0D-8                                           QLDR5510
      IOT = IPT - 1                                                     QLDR5515
      CALL Q1PZSM(DK,DR,DW,RHO,XSTN,A,B,N,QANS,C1CF,IOT)                QLDR5520
```

```
      C1PRIP = -QANS                                                    QLDR5525
      IF(IPT.LE.0) RETURN                                               QLDR5530
      GOTO 60                                                           QLDR5535
   20 Q1PRIP = DCMPLX(0.0D0,0.0D0)                                      QLDR5540
      IF (IPT.LE.0) RETURN                                              QLDR5545
   60 WRITE (6,995) Q1PRIP                                              QLDR5550
  995 FORMAT('0',10X,'C1PRIP = ',E14.7,',',E14.7)                       QLDR5555
      RETURN                                                            QLDR5560
      END                                                               QLDR5565
      SUBROUTINE Q1PZSM (DK,DR,DW,RHO,XSTN,A,B,J,QANS,Q1CF,IPT)         QLDR5570
      IMPLICIT REAL*8(A - E,G,H,M,O,P,R - Y), COMPLEX*16(F,Q,Z)         QLDR5575
      DIMENSION Q1CF(13)                                                QLDR5580
      DIMENSION FV(5),LORR(60),F1T(60),F2T(60),F3T(60),EAT(60),         QLDR5585
     1  ARESTT(60),QESTT(60),EPST(60),QPSUM(60)                         QLDR5590
      F(X) = CLGNDR(X,DR,J,Q1CF)*CDEXP(CEXP*(X))                        QLDR5595
     1     * MMBSJC((OMEGA * DSQRT( (XSTN-X)*(XSTN-X) - YY)), IER)      QLDR5600
      GAMMA = 1.4D0                                                     QLDR5605
      YY = RHO * RHO                                                    QLDR5610
      DM2 = (GAMMA + 1.0D0) * DW                                        QLDR5615
      OMEGA = DSQRT (DK*DK*(1.0D0-DM2)/(DM2*DM2))                       QLDR5620
      CLAMDA = DK/DM2                                                   QLDR5625
      DM = DSQRT(DM2)                                                   QLDR5630
      Q2EXP = CDEXP (DCMPLX (0.0D0, -CLAMDA*XSTN))                      QLDR5635
      QEXP = DCMPLX(0.0D0,CLAMDA)                                       QLDR5640
      U   =  9.0D-13                                                    QLDR5645
      ACC = 1.0D-6                                                      QLDR5650
      IF (IPT.GT.0) WRITE(6,990)DK,DR,DW,RHO,XSTN,A,B,J,IPT             QLDR5655
  990 FORMAT('0',15X,'Q1PZSM ENTERED WITH ARGUMENTS:',/,                QLDR5660
     1' ',15X,'DK',13X,'DR',13X,'DW    ',10X,'RHO',12X,'XSTN',11X,     QLDR5665
     2'A',14X,'B',14X,'J',2X,'IPT',/,                                   QLDR5670
     3' ',15X,7(E14.7,','),I2,2X,I3)                                    QLDR5675
      EFOURU = 4.0*U                                                    QLDR5680
      IFLAG = 1                                                         QLDR5685
      EPS = ACC                                                         QLDR5690
      CERROR = DCMPLX(0.0D0,0.0D0)                                      QLDR5695
      LVL = 1                                                           QLDR5700
      LORR(LVL) = 1                                                     QLDR5705
      QPSUM(LVL) = 0.0                                                  QLDR5710
      ALPHA = A                                                         QLDR5715
      DA = B - A                                                        QLDR5720
      AREA = 0.0                                                        QLDR5725
      AREST = 0.0                                                       QLDR5730
      FV(1) = F(ALPHA)                                                  QLDR5735
      FV(3) = F(ALPHA + 0.5*DA)                                         QLDR5740
      FV(5) = F(ALPHA + DA)                                             QLDR5745
      KOUNT = 3                                                         QLDR5750
      WT = DA/6.0                                                       QLDR5755
      QEST = WT*(FV(1) + 4.0*FV(3) + FV(5))                             QLDR5760
```

```
1     DX = 0.5*DA                                                                    QLDR5765
      FV(2) = F(ALPHA + 0.5*DX)                                                      QLDR5770
      FV(4) = F(ALPHA + 1.5*DX)                                                      QLDR5775
      KOUNT = KOUNT + 2                                                              QLDR5780
      WT = DX/6.0                                                                    QLDR5785
      QESTL = WT*(FV(1) + 4.0*FV(2) + FV(3))                                         QLDR5790
      QESTR = WT*(FV(3) + 4.0*FV(4) + FV(5))                                         QLDR5795
      QSUM = QESTL + QESTR                                                           QLDR5800
      ARESTL = WT*(CDABS(FV(1)) + CDABS(4.0*FV(2)) + CDABS(FV(3)))                   QLDR5805
      ARESTR = WT*(CDABS(FV(3)) + CDABS(4.0*FV(4)) + CDABS(FV(5)))                   QLDR5810
      AREA = AREA + ((ARESTL + ARESTR) - AREST)                                      QLDR5815
      QDIFF = QEST - QSUM                                                            QLDR5820
      IF(CDABS(QDIFF).LE.EPS*CABS(AREA))GO TO 2                                      QLDR5825
      IF(DABS(DX).LE.EFCORU*DABS(ALPHA))GO TO 5                                      QLDR5830
      IF(LVL.GE.60)GO TO 5                                                           QLDR5835
      IF(KOUNT.GE.2000)GO TO 6                                                       QLDR5840
      LVL = LVL + 1                                                                  QLDR5845
      LORR(LVL) = 0                                                                  QLDR5850
      F1T(LVL) = FV(3)                                                               QLDR5855
      F2T(LVL) = FV(4)                                                               QLDR5860
      F3T(LVL) = FV(5)                                                               QLDR5865
      DA = DX                                                                        QLDR5870
      DAT(LVL) = DX                                                                  QLDR5875
      AREST = ARESTL                                                                 QLDR5880
      ARESTT(LVL) = ARESTR                                                           QLDR5885
      QEST = QESTL                                                                   QLDR5890
      QESTT(LVL) = QESTR                                                             QLDR5895
      EPS = EPS/1.4                                                                  QLDR5900
      EPST(LVL) = EPS                                                                QLDR5905
      FV(5) = FV(3)                                                                  QLDR5910
      FV(3) = FV(2)                                                                  QLDR5915
      GO TO 1                                                                        QLDR5920
2     QERROR = QERROR + QDIFF/15.0                                                   QLDR5925
3     IF(LORR(LVL).EQ.0)GO TO 4                                                      QLDR5930
      QSUM = QPSUM(LVL) + QSUM                                                       QLDR5935
      LVL = LVL - 1                                                                  QLDR5940
      IF(LVL.GT.1)GO TO 3                                                            QLDR5945
      QANS = QSUM * Q2EXP / DM                                                       QLDR5950
      IF(IPT.GT.0) GO TO 11                                                          QLDR5955
      IF(IFLAG.EQ.1) RETURN                                                          QLDR5960
      WRITE(6,990) DK,DR,DW,RHO,XSTN,A,E,J,IPT                                       QLDR5965
   11 WRITE(6,995) QANS,IFLAG,IER,QERROR                                             QLDR5970
  995 FORMAT(' ',15X,'RESULTS: QANS = ',E14.7,',',E14.7,/,                           QLDR5975
     1' ',15X,'IFLAG',2X,'IER',5X,'QERROR',/,                                        QLDR5980
     2' ',15X,I3,2X,I3,5X,E14.7,',',E14.7)                                           QLDR5985
      RETURN                                                                         QLDR5990
4     QPSUM(LVL) = QSUM                                                              QLDR5995
      LORR(LVL) = 1                                                                  QLDR6000
```

```
      ALPHA = ALPHA + DA                                                QLDR6005
      DA = DAT(LVL)                                                     QLDR6010
      FV(1) = F1T(LVL)                                                  QLDR6015
      FV(3) = F2T(LVL)                                                  QLDR6020
      FV(5) = F3T(LVL)                                                  QLDR6025
      AREST = ARESTT(LVL)                                               QLDR6030
      QEST = QESTT(LVL)                                                 QLDR6035
      EPS = EPST(LVL)                                                   QLDR6040
      GO TO 1                                                           QLDR6045
    5 IFLAG = 2                                                         QLDR6050
      GO TO 2                                                           QLDR6055
    6 IFLAG = 3                                                         QLDR6060
      GO TO 2                                                           QLDR6065
      END                                                               QLDR6070
      COMPLEX FUNCTION QLGNDR*16(X,DR,N,C1CF)                           QLDR6075
      IMPLICIT REAL * 8 (A-H,O,P,R-Y),  COMPLEX * 16(Q,Z)               QLDR6080
      DIMENSION Q1CF(13)                                                QLDR6085
      ZSUM = 0.0D0                                                      QLDR6090
      DO 10 J1 = 1,N                                                    QLDR6095
      J = J1 - 1                                                        QLDR6100
      ZSUM = ZSUM + PLGNDR(X,DR,J) * C1CF(J1)                           QLDR6105
   10 CONTINUE                                                          QLDR6110
      QLGNDR = ZSUM                                                     QLDR6115
      RETURN                                                            QLDR6120
      END                                                               QLDR6125
      COMPLEX FUNCTION Q1DPHI*16 (DK,DR,DW,RHO,OFFSET,SIGMA,            QLDR6130
     1    N,Q1ABCF,Q1RBCF,XSTN,IPT)                                     QLDR6135
      IMPLICIT REAL*8(A-H,O,P,R-Y), COMPLEX*16(C,Z)                     QLDR6140
      DIMENSION Q1ABCF(13),Q1RBCF(13)                                   QLDR6145
      IF (IPT.GT.0) WRITE(6,990) DK,DR,DW,RHO,OFFSET,SIGMA,XSTN,IPT     QLDR6150
  990 FORMAT('0',10X,'Q1DPHI ENTERED WITH:',/,                          QLDR6155
     1   ' ',10X,'DK',11X,'DR',11X,'DW',11X,'RHO',10X,'OFFSET',7X,'SIGMA'QLDR6160
     2   ,8X,'XSTN',9X,'IPT'/,' ',10X,7(E12.5,',',),I3)                QLDR6165
      RZERO = 0.0D0                                                     QLDR6170
      IOT = IPT - 1                                                     QLDR6175
      Q1DPHI = Q1DREP(DK,DR,DW,RZERO,XSTN,IOT)                          QLDR6180
     1 + Q1DRIP(DK,DR,DW,RZERO,OFFSET,XSTN,Q1RBCF,N,IOT)                QLDR6185
     2 + Q1DAEP(DK,DR,DW,DR,OFFSET,SIGMA,XSTN,IOT)                      QLDR6190
     3 + Q1DAIP(DK,DR,DW,DR,OFFSET,SIGMA,XSTN,Q1ABCF,N,IOT)             QLDR6195
      IF(IPT.LE.0) RETURN                                               QLDR6200
      WRITE (6,995) Q1DPHI                                              QLDR6205
  995 FORMAT('0',10X,'Q1DPHI = ',E14.7,',',E14.7)                       QLDR6210
      RETURN                                                            QLDR6215
      END                                                               QLDR6220
      COMPLEX FUNCTION Q1DREP*16(DK,DR,DW,RHO,XSTN,IPT)                 QLDR6225
      IMPLICIT REAL * 8 (A-H,C,P,R-Y), COMPLEX * 16 (Q,Z)               QLDR6230
      DIMENSION CINP(2)                                                 QLDR6235
      IF (IPT.GT.0) WRITE(6,990) DK,DR,DW,RHO,XSTN,IPT                  QLDR6240
```

```
  990 FORMAT('0',10X,'Q1DRBP ENTERED WITH:',/,                          QLDR6245
     1   ' ',10X,'DK',11X,'DR',11X,'DW',11X,'RHO',10X,'XSTN',9X,'IPT'   QLDR6250
     2   ,/,' ',10X,5(E12.5,','),I3)                                   QLDR6255
      IF(XSTN.LE.RHO-1.0D0) GOTO 20                                     QLDR6260
      ICT = IPT - 1                                                     QLDR6265
      QDK = DCMPLX (0.0D0,DK)                                           QLDR6270
      CALL Q1DWXM(DK,DR,DW,RHO,XSTN,QINP,IOT)                           QLDR6275
      Q1DRBP = -QDK*QINP(2) - QINP(1)                                   QLDR6280
      IF(IPT.LE.0) RETURN                                               QLDR6285
      GOTO 60                                                           QLDR6290
   20 Q1DRBP = DCMPLX(0.0D0,0.0D0)                                      QLDR6295
      IF(IPT.LE.0) RETURN                                               QLDR6300
   60 WRITE (6,995) Q1DRBP                                              QLDR6305
  995 FORMAT('0',10X,'Q1DRBP = ',E14.7,',',E14.7)                       QLDR6310
      RETURN                                                            QLDR6315
      END                                                               QLDR6320
      COMPLEX FUNCTION Q1DABP*16(DK,DR,DW,RHO,OFFSET,SIGMA,XSTN,IPT)    QLDR6325
      IMPLICIT REAL * 8 (A-H,O,P,R-Y), COMPLEX * 16 (Q,Z)               QLDR6330
      DIMENSION QINP(2)                                                 QLDR6335
      IF(IPT.GT.0) WRITE(6,990) DK,DR,DW,RHO,OFFSET,SIGMA,XSTN,IPT      QLDR6340
  990 FORMAT('0',10X,'Q1DABP ENTERED WITH:',/,                          QLDR6345
     1   ' ',10X,'DK',11X,'DR',11X,'DW',11X,'RHO',10X,'OFFSET',7X,      QLDR6350
     2   'SIGMA',8X,'XSTN',9X,'IPT'/,' ',10X,7(E12.5,','),I3)          QLDR6355
      XASTN = XSTN - OFFSET                                             QLDR6360
      IF(XASTN.LE.RHO-1.0D0) GO TO 20                                   QLDR6365
      ICT = IPT - 1                                                     QLDR6370
      QDK = DCMPLX (0.0D0,DK)                                           QLDR6375
      QCONST = CDEXP( DCMPLX(0.0D0,SIGMA))                              QLDR6380
      CALL Q1DWXM(DK,DR,DW,RHO,XASTN,QINP,IOT)                          QLDR6385
      Q1DABP = (QDK*QINP(2) + QINP(1)) * QCONST                         QLDR6390
      IF(IPT.LE.0) RETURN                                               QLDR6395
      GO TO 60                                                          QLDR6400
   20 Q1DABP = DCMPLX(0.0D0,0.0D0)                                      QLDR6405
      IF (IPT.LE.0) RETURN                                              QLDR6410
   60 WRITE (6,995) Q1DABP                                              QLDR6415
  995 FORMAT('0',10X,'Q1DABP = ',E14.7,',',E14.7)                       QLDR6420
      RETURN                                                            QLDR6425
      END                                                               QLDR6430
      SUBROUTINE Q1DWXM(DK,DR,DW,RHO, XSTN,QINP,IPT)                    QLDR6435
C                                                                       QLDR6440
C     N IS THE MAXIMUM DEGREE OF THE TERM U IN THE INTEGRALS            QLDR6445
C                                                                       QLDR6450
      IMPLICIT REAL * 8 (A-H,O,P,R-Y), COMPLEX * 16 (Z,C)               QLDR6455
      DIMENSION QINP(2)                                                 QLDR6460
      IF(IPT.GT.0) WRITE(6,990) DK,DR,RHO,XSTN,IPT                      QLDR6465
  990 FORMAT('0',10X,'Q1DWXM ENTERED WITH ARGUMENTS:',/,                QLDR6470
     1' ',10X,'DK',16X,'DR',16X,'RHO',15X,'XSTN',14X,'IPT',/,           QLDR6475
     2' ',10X,4(E13.6,5X),I2,5X,I3)                                     QLDR6480
```

```
      ICT = IPT - 1                                                     QLDR6485
      B = XSTN - RHO - 1.0D-10                                          QLDR6490
      A = -1.0D0                                                        QLDR6495
      CALL Q1DZXM(DK,DR,DW,RHO,XSTN,A,B,1,QANS,IQT)                     QLDR6500
      QINP(1) = QANS                                                    QLDR6505
      CALL Q1DZXM(DK,DR,DW,RHO,XSTN,A,B,2,QANS,IQT)                     QLDR6510
      QINP(2) = QANS                                                    QLDR6515
      IF(IPT.LE.0) RETURN                                               QLDR6520
      WRITE (6,995)                                                     QLDR6525
  995 FORMAT(' ',1CX,'Q1DWXM RESULTS:',/,                               QLDR6530
     1' ',11X,'J',6X,'QINP(I)')                                         QLDR6535
      DO 1001 I = 1,2                                                   QLDR6540
  996 FORMAT(' ',10X,I3,5X,E12.6,',',E12.6)                             QLDR6545
 1001 WRITE (6,996) I, QINP(I)                                          QLDR6550
      RETURN                                                            QLDR6555
      END                                                               QLDR6560
      SUBROUTINE Q1DZXM (DK,DR,DW,RHO,XSTN,A,B,J,QANS,IFT)              QLDR6565
      IMPLICIT REAL*8(A - E,G,H,M,O,P,R - Y), COMPLEX*16(F,Q,Z)         QLDR6570
      DIMENSION FV(5),LORR(60),F1T(60),F2T(60),F3T(60),DAT(60),         QLDR6575
     1  ARESTT(60),QESTT(60),EPST(60),QPSUM(60)                         QLDR6580
      F(X) = (X**J1)*CDEXP(QEXP*(X))                                    QLDR6585
     1     * MMESJ1((OMEGA * DSQRT( (XSTN-X)*(XSTN-X) - YY)), IER)      QLDR6590
     2     * OMEGA * (XSTN-X)/(DSQRT((XSTN-X)*(XSTN-X)-YY))            QLDR6595
      GAMMA = 1.4D0                                                     QLDR6600
      YY = RHO * RHO                                                    QLDR6605
      DM2 = (GAMMA + 1.0D0) * DW                                        QLDR6610
      OMEGA = DSQRT (DK*DK*(1.0D0-DM2)/(DM2*DM2))                       QLDR6615
      DLAMDA = DK/DM2                                                   QLDR6620
      DM = DSQRT(DM2)                                                   QLDR6625
      QEXP = CDEXP (DCMPLX (0.0D0, -DLAMDA*XSTN))                       QLDR6630
      QEXP = DCMPLX(0.0D0,DLAMDA)                                       QLDR6635
      J1 = J - 1                                                        QLDR6640
      U  =  9.0D-13                                                     QLDR6645
      ACC = 1.0D-6                                                      QLDR6650
      IF (IPT.GT.0) WRITE(6,990)DK,DR,DW,RHO,XSTN,A,B,J,IFT             QLDR6655
  990 FORMAT('0',15X,'Q1DZXM ENTERED WITH ARGUMENTS:',/,                QLDR6660
     1' ',15X,'DK',13X,'DR',13X,'DW    ',10X,'RHO',12X,'XSTN',11X,     QLDR6665
     2'A',14X,'B',14X,'J',2X,'IPT',/,                                   QLDR6670
     3' ',15X,7(E14.7,','),I2,2X,I3)                                    QLDR6675
      EPCURU = 4.0*U                                                    QLDR6680
      IFLAG = 1                                                         QLDR6685
      EPS = ACC                                                         QLDR6690
      QERROR = DCMPLX(0.0D0,0.0D0)                                      QLDR6695
      LVL = 1                                                           QLDR6700
      LORR(LVL) = 1                                                     QLDR6705
      QPSUM(LVL) = 0.0                                                  QLDR6710
      ALPHA = A                                                         QLDR6715
      DA = B - A                                                        QLDR6720
```

```
      AREA = 0.0                                                        QLDR6725
      AREST = 0.0                                                       QLDR6730
      FV(1) = F(ALPHA)                                                  QLDR6735
      FV(3) = F(ALPHA + 0.5*DA)                                         QLDR6740
      FV(5) = F(ALPHA + DA)                                             QLDR6745
      KOUNT = 3                                                         QLDR6750
      WT = DA/6.0                                                       QLDR6755
      QEST = WT*(FV(1) + 4.0*FV(3) + FV(5))                             QLDR6760
    1 DX = 0.5*DA                                                       QLDR6765
      FV(2) = F(ALPHA + 0.5*DX)                                         QLDR6770
      FV(4) = F(ALPHA + 1.5*DX)                                         QLDR6775
      KOUNT = KOUNT + 2                                                 QLDR6780
      WT = DX/6.0                                                       QLDR6785
      QESTL = WT*(FV(1) + 4.0*FV(2) + FV(3))                            QLDR6790
      QESTR = WT*(FV(3) + 4.0*FV(4) + FV(5))                            QLDR6795
      QSUM = QESTL + QESTR                                              QLDR6800
      ARESTL = WT*(CDABS(FV(1)) + CDABS(4.0*FV(2)) + CDABS(FV(3)))      QLDR6805
      ARESTR = WT*(CDABS(FV(3)) + CDABS(4.0*FV(4)) + CDABS(FV(5)))      QLDR6810
      AREA = AREA + ((ARESTL + ARESTR) - AREST)                         QLDR6815
      QDIFF = QEST - QSUM                                               QLDR6820
      IF(CDABS(QDIFF).LE.EPS*DABS(AREA))GO TO 2                         QLDR6825
      IF(DABS(DX).LE.EFCURU*DABS(ALPHA))GO TO 5                         QLDR6830
      IF(LVL.GE.60)GO TO 5                                              QLDR6835
      IF(KOUNT.GE.4000)GO TO 6                                          QLDR6840
      LVL = LVL + 1                                                     QLDR6845
      LORR(LVL) = 0                                                     QLDR6850
      F1T(LVL) = FV(3)                                                  QLDR6855
      F2T(LVL) = FV(4)                                                  QLDR6860
      F3T(LVL) = FV(5)                                                  QLDR6865
      DA = DX                                                           QLDR6870
      DAT(LVL) = DX                                                     QLDR6875
      AREST = ARESTL                                                    QLDR6880
      ARESTT(LVL) = ARESTR                                              QLDR6885
      QEST = QESTL                                                      QLDR6890
      QESTT(LVL) = QESTR                                                QLDR6895
      EPS = EPS/1.4                                                     QLDR6900
      EPST(LVL) = EPS                                                   QLDR6905
      FV(5) = FV(3)                                                     QLDR6910
      FV(3) = FV(2)                                                     QLDR6915
      GO TO 1                                                           QLDR6920
    2 QERROR = QERROR + QDIFF/15.0                                      QLDR6925
    3 IF(LORR(LVL).EQ.0)GO TO 4                                         QLDR6930
      QSUM = QPSUM(LVL) + QSUM                                          QLDR6935
      LVL = LVL - 1                                                     QLDR6940
      IF(LVL.GT.1)GO TO 3                                               QLDR6945
      QANS = ((B**J1)*CDEXP(QEXP*B) - QSLM)*Q2EXP/DN                    QLDR6950
      IF(IPT.GT.0) GO TO 11                                             QLDR6955
      IF(IFLAG.EQ.1) RETURN                                             QLDR6960
```

```
      WRITE(6,990) DK,DF,DW,RHO,XSTN,A,B,J,IPT                          QLDR6965
   11 WRITE(6,995) QANS,IFLAG,IER,QERRCR                                QLDR6970
  995 FORMAT(' ',15X,'Q1DZXM RESULTS: QANS = ',E14.7,',',E14.7,/,       QLDR6975
     1' ',15X,'IFLAG',2X,'IER',5X,'QERRCR',/,                           QLDR6980
     2' ',15X,I3,2X,I3,5X,E14.7,',',E14.7)                              QLDR6985
      RETURN                                                            QLDR6990
    4 QFSUM(LVL) = QSUM                                                 QLDR6995
      LCRR(LVL) = 1                                                     QLDR7000
      ALPHA = ALPHA + CA                                                QLDR7005
      CA = DAT(LVL)                                                     QLDR7010
      FV(1) = F1T(LVL)                                                  QLDR7015
      FV(3) = F2T(LVL)                                                  QLDR7020
      FV(5) = F3T(LVL)                                                  QLDR7025
      AREST = ARESTT(LVL)                                               QLDR7030
      QEST = QESTT(LVL)                                                 QLDR7035
      EPS = EPST(LVL)                                                   QLDR7040
      GO TO 1                                                           QLDR7045
    5 IFLAG = 2                                                         QLDR7050
      GO TO 2                                                           QLDR7055
    6 IFLAG = 3                                                         QLDR7060
      GO TO 2                                                           QLDR7065
      END                                                               QLDR7070
      COMPLEX FUNCTION Q1DAIP*16(DK,DR,DW,RHO,OFST,SGMA,XSTN,Q1CF,N,IPT)QLDR7075
      IMPLICIT REAL*8 (A-H,O,P,R-Y), COMPLEX * 16 (Q,Z)                 QLDR7080
      DIMENSION Q1CF(13)                                                QLDR7085
      IF (IPT.GT.0) WRITE(6,990) DK,DR,DW,RHO,OFST,SGMA,XSTN,IPT        QLDR7090
  990 FORMAT('0',10X,'Q1DAIP ENTERED WITH:',/,                          QLDR7095
     1  ' ',10X,'DK',11X,'DR',11X,'DW',11X,'RHO',10X,'OFST',9X,'SGMA',9XQLDR7100
     2  ,'XSTN',9X,'IPT'/,' ',10X,7(E12.5,','),I3)                     QLDR7105
      XASTN = XSTN - OFST                                               QLDR7110
      IF(XASTN.LE.DR+RHO-1.0D0) GO TO 20                                QLDR7115
      IOT = IPT - 1                                                     QLDR7120
      QCONST = QCEXP( DCMPLX(0.0D0,SGMA))                               QLDR7125
      A = DR-1.0D0                                                      QLDR7130
      B = XASTN - RHO - 1.0D-8                                          QLDR7135
      CALL Q1DZSM(DK,DR,DW,RHO,XSTN,A,B,N,QANS,Q1CF,IOT)                QLDR7140
      Q1DAIP = QANS                                                     QLDR7145
      IF(IPT.LE.0) RETURN                                               QLDR7150
      GOTO 60                                                           QLDR7155
   20 Q1DAIP = DCMPLX(0.0D0,0.0D0)                                      QLDR7160
      IF(IPT.LE.0)RETURN                                                QLDR7165
   60 WRITE (6,995) Q1DAIP                                              QLDR7170
  995 FORMAT('0',10X,'Q1DAIP = ',E14.7,',',E14.7)                       QLDR7175
      RETURN                                                            QLDR7180
      END                                                               QLDR7185
      COMPLEX FUNCTION Q1DRIP*16(DK,DF,DW,RHO,OFFSET,XSTN,Q1CF,N,IPT)   QLDR7190
      IMPLICIT REAL*8 (A-H,O,P,R-Y), COMPLEX * 16 (Q,Z)                 QLDR7195
      DIMENSION Q1CF(13)                                                QLDR7200
```

```
      IF (IPT.GT.0) WRITE(6,990) DK,DR,DW,RHO,XSTN,IPT                  QLDR7205
  990 FORMAT('0',10X,'Q1DRIP ENTERED WITH:',/,                          QLDR7210
     1   ' ',10X,'DK',11X,'DR',11X,'DW',11X,'RHO',10X,                 QLDR7215
     2   'XSTN',9X,'IPT'/,' ',10X,5(E12.5,','),I3)                     QLDR7220
      IF(XSTN.LE.DR+RHO+OFFSET-1.0D0) GOTO 20                           QLDR7225
      ICT = IPT - 1                                                     QLDR7230
      A = DR-1.0D0                                                      QLDR7235
      B = XSTN - RHO - 1.0D-8                                           QLDR7240
      CALL Q1DZSM(DK,DR,DW,RHO,XSTN,A,B,N,QANS,Q1CF,IOT)                QLDR7245
      Q1DRIP = -QANS                                                    QLDR7250
      IF(IPT.LE.0) RETURN                                               QLDR7255
      GOTO 60                                                           QLDR7260
   20 Q1DRIP = DCMPLX(0.0D0,0.0D0)                                      QLDR7265
      IF (IPT.LE.0) RETURN                                              QLDR7270
   60 WRITE (6,995) Q1DRIP                                              QLDR7275
  995 FORMAT('0',10X,'Q1DRIP = ',E14.7,',',E14.7)                       QLDR7280
      RETURN                                                            QLDR7285
      END                                                               QLDR7290
      SUBROUTINE Q1DZSM (DK,DR,DW,RHO,XSTN,A,B,J,QANS,Q1CF,IPT)         QLDR7295
      IMPLICIT REAL*8(A - E,G,H,M,C,P,R - Y), COMPLEX*16(F,Q,Z)         QLDR7300
      DIMENSION Q1CF(13)                                                QLDR7305
      DIMENSION FV(5),LCRR(60),F1T(60),F2T(60),F3T(60),DAT(60),         QLDR7310
     1  ARESTT(60),QESTT(60),EPST(60),QPSUM(60)                         QLDR7315
      F(X) = QLGNDR(X,DR,J,Q1CF)*CDEXP(QEXP*(X))                        QLDR7320
     1      * MMBSJ1((OMEGA * DSQRT( (XSTN-X)*(XSTN-X) - YY)), IER)     QLDR7325
     2      * OMEGA * (XSTN-X)/(DSQRT((XSTN-X)*(XSTN-X)-YY))           QLDR7330
      GAMMA = 1.4D0                                                     QLDR7335
      YY = RHO * RHO                                                    QLDR7340
      DM2 = (GAMMA + 1.0D0) * DW                                        QLDR7345
      OMEGA = DSQRT (DK*DK*(1.0D0-DM2)/(DM2*DM2))                       QLDR7350
      CLAMDA = DK/DM2                                                   QLDR7355
      DM = DSQRT(DM2)                                                   QLDR7360
      Q2EXP = CDEXP (DCMPLX (0.0D0, -CLAMDA*XSTN))                      QLDR7365
      QEXP = DCMPLX(0.0D0,CLAMDA)                                       QLDR7370
      U   =  9.0D-13                                                    QLDR7375
      ACC = 1.0D-6                                                      QLDR7380
      IF (IPT.GT.0) WRITE(6,990)DK,DR,DW,RHO,XSTN,A,B,J,IPT             QLDR7385
  990 FORMAT('0',15X,'Q1DZSM ENTERED WITH ARGUMENTS:',/,                QLDR7390
     1' ',15X,'DK',13X,'DR',13X,'DW    ',10X,'RHO',12X,'XSTN',11X,     QLDR7395
     2'A',14X,'B',14X,'J',2X,'IPT',/,                                   QLDR7400
     3' ',15X,7(E14.7,','),I2,2X,I3)                                    QLDR7405
      EFCURU = 4.0*U                                                    QLDR7410
      IFLAG = 1                                                         QLDR7415
      EPS = ACC                                                         QLDR7420
      CERROR = DCMPLX(0.0D0,0.0D0)                                      QLDR7425
      LVL = 1                                                           QLDR7430
      LCRR(LVL) = 1                                                     QLDR7435
      QPSUM(LVL) = 0.0                                                  QLDR7440
```

```
      ALPHA = A                                                                 QLDR7445
      DA = B - A                                                                QLDR7450
      AREA = 0.0                                                                QLDR7455
      ARFST = 0.0                                                               QLDR7460
      FV(1) = F(ALPHA)                                                          QLDR7465
      FV(3) = F(ALPHA + 0.5*DA)                                                 QLDR7470
      FV(5) = F(ALPHA + DA)                                                     QLDR7475
      KCLNT = 3                                                                 QLDR7480
      WT = DA/6.0                                                               QLDR7485
      GEST = WT*(FV(1) + 4.0*FV(3) + FV(5))                                     QLDR7490
    1 DX = 0.5*DA                                                               QLDR7495
      FV(2) = F(ALPHA + C.5*DX)                                                 QLDR7500
      FV(4) = F(ALPHA + 1.5*DX)                                                 QLDR7505
      KCLNT = KOUNT + 2                                                         QLDR7510
      WT = DX/6.0                                                               QLDR7515
      QESTL = WT*(FV(1) + 4.0*FV(2) + FV(3))                                    QLDR7520
      GESTR = WT*(FV(3) + 4.0*FV(4) + FV(5))                                    QLDR7525
      QSUM = QESTL + GESTR                                                      QLDR753C
      ARESTL = WT*(CDABS(FV(1)) + CDABS(4.0*FV(2)) + CDABS(FV(3)))              QLDR7535
      ARESTR = WT*(CDABS(FV(3)) + CDABS(4.0*FV(4)) + CDABS(FV(5)))              QLDR7540
      AREA = AREA + ((ARESTL + ARESTR) - ARFST)                                 QLDR7545
      QDIFF = QEST - QSUM                                                       QLDR7550
      IF(CDABS(QDIFF).LE.EPS*DABS(AREA))GO TO 2                                 QLDR7555
      IF(DABS(DX).LE.EFCURU*DABS(ALPHA))GO TO 5                                 QLDR756C
      IF(LVL.GE.60)GO TO 5                                                      QLDR7565
      IF(KCUNT.GE.4000)GO TO 6                                                  QLDR7570
      LVL = LVL + 1                                                             QLDR7575
      LCRR(LVL) = 0                                                             QLDR7580
      F1T(LVL) = FV(3)                                                          QLDR7585
      F2T(LVL) = FV(4)                                                          QLDR7590
      F3T(LVL) = FV(5)                                                          QLDR7595
      DA = DX                                                                   QLDR760C
      DAT(LVL) = DX                                                             QLDR7605
      AFEST = ARESTL                                                            QLDR7610
      ARESTT(LVL) = ARESTR                                                      QLDR7615
      GEST = GESTL                                                              QLDR7620
      GESTT(LVL) = GESTR                                                        QLDR7625
      EPS = EPS/1.4                                                             QLDR7630
      EPST(LVL) = EPS                                                           QLDR7635
      FV(5) = FV(3)                                                             QLDR764C
      FV(3) = FV(2)                                                             QLDR7645
      GO TO 1                                                                   QLDR7650
    2 GERRCR = GERROR + GDIFF/15.0                                              QLDR7655
    3 IF(LCRR(LVL).EQ.0)GO TO 4                                                 QLDR7660
      QSUM = QPSUM(LVL) + QSUM                                                  QLDR7665
      LVL = LVL - 1                                                             QLDR7670
      IF(LVL.GT.1)GO TO 3                                                       QLDR7675
      QANS = (QLGNDR(B,DR,J,Q1CF)*CDEXP(CEXP*B) - QSUM)*Q2EXP/DM                QLDR7680
```

```
      IF(IPT.GT.0) GO TO 11                                              QLDR7685
      IF(IFLAG.EQ.1) RETURN                                              QLDR7690
      WRITE(6,990) DK,DR,DW,RHO,XSTN,A,E,J,IPT                           QLDR7695
   11 WRITE(6,995) QANS,IFLAG,IER,CERRCR                                 QLDR7700
  995 FORMAT(' ',15X,'CIDZSM RESULTS: CANS = ',E14.7,',',E14.7,/,        QLDR7705
     1' ',15X,'IFLAG',2X,'IER',5X,'CERRCR',/,                            QLDR7710
     2' ',15X,I3,2X,I3,5X,E14.7,',',E14.7)                               QLDR7715
      RETURN                                                             QLDR7720
    4 QFSUM(LVL) = QSUM                                                  QLDR7725
      LCRR(LVL) = 1                                                      QLDR7730
      ALPHA = ALPHA + DA                                                 QLDR7735
      DA = DAT(LVL)                                                      QLDR7740
      FV(1) = F1T(LVL)                                                   QLDR7745
      FV(3) = F2T(LVL)                                                   QLDR7750
      FV(5) = F3T(LVL)                                                   QLDR7755
      AREST = ARESTT(LVL)                                                QLDR7760
      QEST = QESTT(LVL)                                                  QLDR7765
      EPS = EPST(LVL)                                                    QLDR7770
      GO TO 1                                                            QLDR7775
    5 IFLAG = 2                                                          QLDR7780
      GO TO 2                                                            QLDR7785
    6 IFLAG = 3                                                          QLDR7790
      GO TO 2                                                            QLDR7795
      END                                                                QLDR7800
```

7. Gorelov Spanning Function.

The subprogram for function used by Gorelov is shown below.

```
      REAL FUNCTION PLGNDR*8(X,DR,N)
      IMPLICIT REAL*8(A-H,O-Z)
      IF(N.EQ.0) GOTO 100
      ETA = DARCOS(-X)
      ETASTR = DARCOS(1.0D0 - DR)
      FN = DFLOAT(N)
      PLGNDR = DCOS(FN*ETA)-DCOS(FN*ETASTR)
      RETURN
  100 PLGNDR =  1.0D0
      RETURN
      END
```

8. Linear Expansion Program

The program based on the linear expansion for small k is shown below.

```
      IMPLICIT REAL*8(A-H,O,P,R-Y), COMPLEX*16(Q,Z)                      LIN00010
      DIMENSION X(13), Q2PT(13), Q2CP(13)                               LIN00020
C     BLOCK ONE   READ AND EDIT DATA                                    LIN00030
      CALL ERRSET(208,256,-1,1)                                         LIN00040
C     WRITE (6,910)                                                     LIN00050
    1 CALL READ(DK,DR,DW,RHO,OFFSET,SIGMA,N,NF,IPT)                     LIN00060
C     IF(IPT.GT.0)WRITE(6,990)DK,DR,DW,SIGMA,OFFSET,RHO,N,NF            LIN00070
  910 FORMAT ('1',5X,'GORELOV CASCADE PROGRAM LINEARIZED FOR SMALL DK')  LIN00080
  990 FORMAT('0',5X,'INPUT VALUES TRANSMITTED TO MAIN PROGRAM:',/,      LIN00090
     1' ',10X,'DK',13X,'DR',13X,'DW    ',10X,'SIGMA',10X,'OFFSET',/,  LIN00100
     1' ',10X,5(E12.5,3X),/,                                            LIN00110
     2'0',10X,'RHO',12X,'N',5X,'NFCN'                                   LIN00120
     3,/,' ',10X,1(E12.5,3X), I2,4X,I2)                                 LIN00130
      CALL ABSA(N,OFFSET,X,RHO,DW    )                                  LIN00140
C                                                                       LIN00150
CC                                                                      LIN00160
C     BLOCK TWO    CALLS CALCULATION ROUTINES FOR ZONE ONE AND ZONE TWO  LIN00170
      CALL Q$ONE(DK,DR,DW,RHO,OFFSET,SIGMA,N,NF,IPT,X)                  LIN00180
      GO TO 1                                                           LIN00190
      END                                                               LIN00200
      SUBROUTINE READ (DK,DR,DW,RHO,OFFSET,SIGMA,N,NF,IPT)              LIN00210
      IMPLICIT REAL*8(A-H,O,P,R-Y)                                      LIN00220
      READ(5,905) DK,DR,DW,RHO,OFFSET,SIGMA,N,NF,IPT                    LIN00230
      IF (DK.LT.0.0D0) STOP                                             LIN00240
      IF (IPT.GT.0) WRITE (6,910)                                       LIN00250
      GAMMA = 1.4D0                                                     LIN00260
      DR = RHO * DSQRT((GAMMA + 1.0D0) * DW)                            LIN00270
      RHO = DR                                                          LIN00280
      IF(N.LE.2) GO TO  1                                               LIN00290
      N = 2                                                             LIN00300
C     WRITE(6,935 )                                                     LIN00310
    1 IF(OFFSET.LE.1.9D0) GO TO 2                                       LIN00320
      WRITE(6,940)                                                      LIN00330
      OFFSET = OFFSET-1.0D0                                             LIN00340
      GO TO 1                                                           LIN00350
    2 CONTINUE                                                          LIN00360
C     IF (IPT.GT.0) WRITE (6,925) DK,DR,DW,SIGMA,OFFSET,RHO,N,NF,IPT    LIN00370
      RETURN                                                            LIN00380
  905 FORMAT(6F10.4,3I2)                                                LIN00390
  910 FORMAT ('1',10X,'GORELOV SLIGHTLY SUPERSONIC CASCADE PROGRAM',/,  LIN00400
     *     ' ',15X,'LINEARIZED FOR SMALL FREQUENCY, DK.')               LIN00410
  925 FORMAT ('0',10X,'DK',13X,'DR',13X,'DW    ',10X,'SIGMA',10X,'OFFSET'LIN00420
     1,/,' ',10X,5(E12.5,3X),/,                                         LIN00430
     2'0',10X,'RHO',12X,'N',5X,'NFCN',2X,'IPT'                          LIN00440
     3,/,' ',10X,1(E12.5,3X), I2,4X,I2,4X,I2)                           LIN00450
  935 FORMAT('0',5X,'ORIGINAL N TOO LARGE (.GT.13) - RESET TO N = 13')  LIN00460
  940 FORMAT('0',5X,'OFFSET TOO LARGE (.GT.1.9), RESET AS OFFSET =      LIN00470
     1OFFSET - 1.0D0')                                                  LIN00480
```

```
      END                                                                LINOC49C
      SUBROUTINE ABSA (N,OFFSET,X,DR,DW   )                              LINO0500
      IMPLICIT REAL*8 (A-H,O,P,R-Y), COMPLEX*16(Q,Z)                     LINO0510
      DIMENSION X(13)                                                    LINO0520
      XINT =  (2.0D0-DR)/DFLOAT(N+1)                                     LINO0530
      XL  = DR - 1.0D0 + 1.0D-8                                          LINOC540
      DO 10 I = 1,N                                                      LINO0550
      X(I) = XL + (XINT*DFLOAT(I))                                       LINO0560
      IF(X(I).EQ.0.0D0) X(I) = .1D-14                                    LINO0570
   10 CONTINUE                                                           LINO058C
      RETURN                                                             LINO0590
      END                                                                LINOC60C
      SUBROUTINE Q$ONE(DK,DR,DW,RHO,OFFSET,SIGMA,N,NF,IPT,X)             LINO0610
      IMPLICIT REAL*8(A-H,O,P,R-Y), COMPLEX*16(Z,Q)                      LINO0620
      DIMENSION Q1COF(26), Q1INT(26,26), Q1RBP(13), Q1ABP(13)            LINOC63C
      DIMENSION Q1RBCF(13), Q1ABCF(13)                                   LINO0640
      DIMENSION X(13)                                                    LINO0650
C     IF(IPT.GT.0)WRITE(6,990)DK,DR,RHO,OFFSET,SIGMA,N,NF,IPT            LINO066C
      ICT = IPT - 1                                                      LINO0670
  990 FORMAT('0  SUBROUTINE Q$ONE ENTERED WITH:',/,                      LINO0680
     1' ',5X,'DK',13X,'DR',13X,'RHO',12X,'OFFSET',9X,'SIGMA',10X,        LINO0690
     2'N',3X,'NF',2X,'IPT',/,                                            LINO0700
     3  5X,5(E12.5,3X),I2,2X,I2,2X,I3)                                   LINOC71C
      CALL Q1ZINT(DK,DR,DW,RHO,OFFSET,SIGMA,N,X,ICT,Q1COF,Q1INT)         LINO0720
      CALL Q1COEF(Q1COF,Q1INT,N,IOT,Q1ABCF,Q1RBCF)                       LINOC73C
      CALL Q1PPOT(DK,DR,DW,RHO,OFFSET,SIGMA,N,X,Q1ABCF,Q1RBCF,ICT)       LINOC740
      IOT = 0                                                            LINO0750
      CALL Q1DCOF(DK,DR,DW,RHO,OFFSET,SIGMA,N,Q1ABCF,Q1RBCF,IOT)         LINO0760
      RETURN                                                             LINO0770
      END                                                                LINO0780
C                                                                        LINOC790
CC                                                                       LINO0800
C                                                                        LINO0810
      SUBROUTINE Q1ZINT(DK,DR,DW,RHO,OFFSET,SIGMA,N,X,                   LINO0820
     1      IPT,Q1COF,Q1INT)                                             LINO0830
      IMPLICIT REAL * 8(A-H,O,P,R-Y), COMPLEX * 16(Z,Q)                  LINO0840
      DIMENSION Q1COF(26), Q1INT(26,26), QINTRP(13), QINTAP(13)          LINOC85C
      DIMENSION  X(13)                                                   LINO0860
  990 FORMAT ('0',10X,'Q1ZINT ENTERED WITH:',/,                          LINO0870
     1 ' ',10X,'DK',13X,'DR',13X,'RHO',12X,'OFFSET',9X,'SIGMA',10X,      LINO0880
     2  'N',2X,'IPT',/,                                                 LINO0890
     3  ' ',10X,5(E12.5,3X),I2,2X,I3)                                   LINO0900
      CCONST = CDEXP(DCMPLX(0.0D0,SIGMA))                                LINOC91C
      IF (IPT.GT.0) WRITE (6,990) DK,DR,RHO,OFFSET,SIGMA,N,IPT           LINO0920
      ICT = IPT - 1                                                      LINOC93C
      GAMMA = 1.4D0                                                      LINO0940
      CLAMDA = DK/((GAMMA+1.0D0)*DW)                                     LINO0950
      DO 90 I = 1,N                                                      LINO0960
```

```
      IN = I + N                                                        LINCC97C
      XSTN = X(I)                                                       LINCC98C
      XC = XSTN - DR                                                    LINCC990
      CEXP = CDEXP(DCMPLX(C.0D0,DLAMDA*XSTN))                           LINC1000
      CALL Q1CAIP(DK,DR,DW,RHO,OFFSET,SIGMA,XSTN,CINTAP,N,IOT)          LINC1010
      CALL Q1CRIP(DK,DR,DW,RHO,OFFSET,XSTN,QINTRP,N,ICT)                LINC1020
      DO 20 J = 1,N                                                     LINC1030
       JN = N + J                                                       LINC1C4C
      J1 = J-1                                                          LINC1050
      Q1INT(I,1) = DCMPLX(1.0D0,DLAMDA*XSTN)                            LINC1C6C
      TEMP = DR-XSTN-1.0D0                                              LINC1070
      Q1INT(I,2) = DCMPLX(TEMP,DLAMDA*XSTN*TEMP)                        LINC1C8C
      Q1INT(IN,J) = QINTRP(J)                                           LINC1C9C
      Q1INT(I,JN) =  QINTAP(J)                                          LINC1100
      XASTN = XSTN - OFFSET                                             LINC111C
      Q1INT(IN,3) = DCMPLX(1.0D0,DLAMDA*XASTN)                          LINC1120
      TEMP = DR-XASTN-1.0D0                                             LINC113C
      Q1INT(IN,4) = DCMPLX(TEMP,DLAMDA*XASTN*TEMP)                      LINC1140
   20 CONTINUE                                                          LINC1150
      Q1COF(IN) = -Q1CRBP(DK,DR,DW,RHO,XSTN,IOT)                        LINC1160
      Q1COF(I)  = -Q1CABP(DK,DR,DW,RHO,OFFSET,SIGMA,XSTN,IOT)           LINC1170
   90 CONTINUE                                                          LINC1180
      RETURN                                                            LINC1190
      END                                                               LINC12CC
      COMPLEX FUNCTION Q1CRBP*16(DK,DR,DW,RHO,XSTN,IPT)                 LINC1210
      IMPLICIT REAL * 8 (A-H,O,P,R-Y), COMPLEX * 16 (Q,Z)               LINC1220
      DIMENSION QINP(2)                                                 LINC1230
      IF (IPT.GT.0) WRITE(6,990) DK,DR,DW,RHO,XSTN,IPT                  LINC124C
  990 FORMAT('0',10X,'Q1CRBP ENTERED WITH:',/,                          LINC1250
     1   ' ',10X,'DK',11X,'DR',11X,'DW',11X,'RHO',10X,'XSTN',9X,'IPT'   LINC1260
     2   ,/,' ',10X,5(E12.5,','),I3)                                   LINC1270
      IF(XSTN.LE.RHO-1.0D0) GOTO 20                                     LINC1280
      IF(XSTN.GT.2.0D0) GOTO 20                                         LINC1290
      ICT = IPT - 1                                                     LINC1300
      QDK = DCMPLX (0.0D0,DK)                                           LINC1310
      GAMMA = 1.4D0                                                     LINC1320
      DLAMDA = DK/((GAMMA+1.0D0)*DW)                                    LINC1330
      Q1CRBP = DCMPLX(1.0D0,(DK+DLAMDA)*(XSTN-DR))                      LINC1340
      IF(IPT.LE.0) RETURN                                               LINC1350
      GOTO 30                                                           LINC1360
   20 Q1CRBP = DCMPLX(0.0D0,0.0D0)                                      LINC1370
      IF(IPT.LE.0) RETURN                                               LINC1380
   30 WRITE (6,995) Q1CRBP                                              LINC1390
  995 FORMAT('C',10X,'Q1CRBP = ',E14.7,',',E14.7)                       LINC1400
      RETURN                                                            LINC141C
      END                                                               LINC1420
      COMPLEX FUNCTION Q1CABP*16(DK,DR,DW,RHO,OFFSET,SIGMA,XSTN,IPT)    LINC1430
      IMPLICIT REAL * 8 (A-H,O,P,R-Y), COMPLEX * 16 (Q,Z)               LINC1440
```

```
      DIMENSION QINP(2)                                                 LIN01450
      IF(IPT.GT.0) WRITE(6,990) DK,DR,DW,RHO,OFFSET,SIGMA,XSTN,IPT      LIN01460
  990 FORMAT('0',10X,'Q1CABP ENTERED WITH:',/,                          LIN01470
     1   ' ',10X,'DK',11X,'DR',11X,'DW',11X,'RHO',10X,'OFFSET',7X,     LIN01480
     2   'SIGMA',8X,'XSTN',9X,'IPT'/,' ',10X,7(E12.5,','),I3)          LIN01490
      XASTN = XSTN - OFFSET                                             LIN01500
      IF(XASTN.LE.RHO-1.0D0) GOTO 20                                    LIN01510
      IF(XASTN.GT.2.0D0) GOTO 20                                        LIN01520
      IOT = IPT - 1                                                     LIN01530
      CDK = DCMPLX (0.0D0,DK)                                           LIN01540
      CCONST = CDEXP( DCMPLX(0.0D0,SIGMA))                              LIN01550
      GAMMA = 1.4D0                                                     LIN01560
      DLAMDA = DK/((GAMMA+1.0D0)*DW)                                    LIN01570
      Q1CABP = DCMPLX(1.0D0,(DK+DLAMDA)*(XASTN-DR))*QCONST              LIN01580
      IF(IPT.LE.0) RETURN                                               LIN01590
      GO TO 30                                                          LIN01600
   20 Q1CABP = DCMPLX(0.0D0,0.0D0)                                      LIN01610
      IF (IPT.LE.0) RETURN                                              LIN01620
   30 WRITE (6,995) Q1CABP                                              LIN01630
  995 FORMAT ('0',10X,'Q1CABP = ',E14.7,',',E14.7)                      LIN01640
      RETURN                                                            LIN01650
      END                                                               LIN01660
      SUBROUTINE Q1CAIP(DK,DR,DW,RHO,OFST,SGMA,XSTN,QCAIP,N,IPT)        LIN01670
      IMPLICIT REAL*8 (A-H,O,P,R-Y), COMPLEX * 16 (Q,Z)                 LIN01680
      DIMENSION  QINP(13), QCAIP(13)                                    LIN01690
      IF (IPT.GT.0) WRITE(6,990) DK,DR,DW,RHO,OFST,XSTN,IPT             LIN01700
  990 FORMAT('0',10X,'Q1CAIP ENTERED WITH:',/,                          LIN01710
     1   ' ',10X,'DK',11X,'DR',11X,'DW',11X,'RHO',10X,'OFST',9X,'SGMA',9XLIN01720
     2   ,'XSTN',9X,'IPT'/,' ',10X,7(E12.5,','),I3)                    LIN01730
      XASTN = XSTN - OFST                                               LIN01740
      IF(XASTN.LE.DR+RHO-1.0D0) GO TO 20                                LIN01750
      IF(XASTN.GT.2.0D0) GOTO 20                                        LIN01760
      IOT = IPT - 1                                                     LIN01770
      QCONST = CDEXP( DCMPLX(0.0D0,SGMA))                               LIN01780
      GAMMA = 1.4D0                                                     LIN01790
      DLAMDA = DK/((GAMMA+1.0D0)*DW)                                    LIN01800
      QCAIP(1) = DCMPLX(1.0D0,DLAMDA*(XASTN-RHO))                       LIN01810
      TEMP = DR+RHO-XASTN-1.0D0                                         LIN01820
      QCAIP(2) = DCMPLX(TEMP,DLAMDA*(XASTN-RHO)*TEMP)                   LIN01830
      IF(IPT.LE.0) RETURN                                               LIN01840
      GOTO 30                                                           LIN01850
   20 ZERO = DCMPLX(0.0D0,0.0D0)                                        LIN01860
      DO 25 I = 1,N                                                     LIN01870
      QCAIP(I) = ZERO                                                   LIN01880
   25 CONTINUE                                                          LIN01890
      IF(IPT.LE.0)RETURN                                                LIN01900
   30 WRITE (6,995)                                                     LIN01910
  995 FORMAT ('0',10X,'Q1CAIP RESULTS: J     QCAIP(J)')                LIN01920
```

```
      DO 40 J = 1,N                                                          LINO1930
      WRITE (6,996) J,QCAIP(J)                                               LINO1940
  996 FORMAT(' ',26X,I2,3X,E14.7,',',E14.7)                                  LINO1950
   40 CONTINUE                                                               LINO1960
      RETURN                                                                 LINO1970
      END                                                                    LINO1980
      SUBROUTINE  Q1CRIP (DK,DR,DW,RHO,CFFSET,XSTN,QCRIP,N,IPT)              LINO1990
      IMPLICIT REAL*8 (A-H,C,P,R-Y), COMPLEX * 16 (Q,Z)                      LINO2000
      DIMENSION  QCRIP(13)                                                   LINO2010
      IF (IPT.GT.0) WRITE(6,990) DK,DR,DW,RHO,XSTN,IPT                       LINO2020
  990 FORMAT('0',10X,'Q1CRIP ENTERED WITH:',/,                               LINO2030
     1   ' ',10X,'DK',11X,'DR',11X,'DW',11X,'RHO',10X,                       LINO2040
     2   'XSTN',9X,'IPT'/,' ',10X,5(E12.5,','),I3)                           LINO2050
      IF(XSTN.LE.DR+RHO+CFFSET-1.0D0) GOTO 20                                LINO2060
      IF(XSTN.GT.2.0D0) GOTO 20                                              LINO2070
      IDT = IPT - 1                                                          LINO2080
      GAMMA = 1.4D0                                                          LINO2090
      DLAMDA = DK/((GAMMA+1.0D0)*DW)                                         LINO2100
      QCRIP(1) = DCMPLX(1.0D0,DLAMDA*(XSTN-RHO))                             LINO2110
      TEMP = DR+RHO-XSTN-1.0D0                                               LINO2120
      QCRIP(2)  = DCMPLX(TEMP,DLAMDA*(XSTN-RHO)*TEMP)                        LINO2130
      IF(IPT.LE.0) RETURN                                                    LINO2140
      GOTO 30                                                                LINO2150
   20 ZERO = DCMPLX(0.0D0,0.0D0)                                             LINO2160
      DO 25 I = 1,N                                                          LINO2170
   25 QCRIP(I) = ZERO                                                        LINO2180
      IF (IPT.LE.0) RETURN                                                   LINO2190
   30 WRITE (6,995)                                                          LINO2200
  995 FORMAT ('0',10X,'Q1CRIP RESULTS: J     QCRIP(J)')                     LINO2210
      DO 40 J = 1,N                                                          LINO2220
      WRITE (6,996) J,QCRIP(J)                                               LINO2230
  996 FORMAT(' ',26X,I2,3X,E14.7,',',E14.7)                                  LINO2240
   40 CONTINUE                                                               LINO2250
      RETURN                                                                 LINO2260
      END                                                                    LINO2270
      SUBROUTINE Q1COEF( Q1COF,Q1INT,N,IPT,Q1ABCF,Q1RBCF)                    LINO2280
      IMPLICIT REAL * 8(A-H,O,P,R-Y), COMPLEX * 16 (Z,Q)                     LINO2290
      DIMENSION Q1COF(26), Q1INT(26,26), ZWA(300)                            LINO2300
      DIMENSION Q1ABCF(13),Q1RBCF(13)                                        LINO2310
      IF (IPT.GE.0) WRITE (6,90)                                             LINO2320
      IB = 26                                                                LINO2330
      M = 1                                                                  LINO2340
      N2 = 2*N                                                               LINO2350
      IF (IPT.LE.0) GO TO 5                                                  LINO2360
      WRITE (6,98) N,N2                                                      LINO2370
   98 FORMAT('0',5X,'Q1COEF ENTERED WITH ',I2,' DEG  PWR SERIES (',I2,       LINO2380
     1  ' SQUARE MATRIX)')                                                   LINO2390
      DO 2 I = 1,N2                                                          LINO2400
```

```
      WRITE(6,92) I, I, C1COF(I)                                        LIN02410
   92 FORMAT('0',10X,'Q1COEF EQUATION SYSTEM, ROW ',I2,/,               LIN02420
     1' ',10X,'Q1COF(',I2,') = ',E14.7,',',E14.7)                       LIN02430
      DO 2 J = 1,N                                                      LIN02440
      J2=J+N                                                            LIN02450
      WRITE(6,91) I,J,C1INT(I,J),I,J2,C1INT(I,J2)                       LIN02460
   91 FORMAT(' ',15X,2('QINT(',I2,',',I2,') = ',E14.7,',',E14.7,10X))   LIN02470
    2 CONTINUE                                                          LIN02480
    5 IA = 26                                                           LIN02490
      IJOB=0                                                            LIN02500
      CALL LEQT1C(Q1INT,N2,IA,Q1COF,M,IE,IJOB,ZWA,IER)                  LIN02510
      IF(IER.EQ.0) GOTO 30                                              LIN02520
      IF (IER.EQ.129) GO TO 10                                          LIN02530
      WRITE(6,93)                                                       LIN02540
   93 FORMAT( 'C',10X,'C1COEF - ITERATIVE IMPROVEMENT FAILED, MATRIX TOOLIN02550
     1 ILL-CONDITIONED. USE RESULTS WITH CAUTION.')                     LIN02560
      GO TO 30                                                          LIN02570
   10 WRITE (6,95)                                                      LIN02580
   95 FORMAT('0',10X,'Q1COEF - MATRIX ALGORITHMICALLY SINGULAR.  COEFFICLIN02590
     1IENTS SET TO ZERO.')                                              LIN02600
      ZERO = CMPLX(0.0D0,0.0D0)                                         LIN02610
      DO 20 I = 1,N                                                     LIN02620
      I2 = I+N                                                          LIN02630
      Q1COF(I) = ZERO                                                   LIN02640
      Q1COF(I2) = ZERO                                                  LIN02650
   20 CONTINUE                                                          LIN02660
   30 CONTINUE                                                          LIN02670
      DO 35 I = 1,N                                                     LIN02680
      IN = I + N                                                        LIN02690
      Q1ABCF(I) = Q1COF(IN)                                             LIN02700
      Q1RBCF(I) = Q1COF(I)                                              LIN02710
   35 CONTINUE                                                          LIN02720
      IF(IPT.LE.0) RETURN                                               LIN02730
      IF (IPT.GE.2) WRITE (6,94)                                        LIN02740
      DO 40 I = 1,N                                                     LIN02750
      IM1 = I-1                                                         LIN02760
      WRITE (6,99) I, IM1, Q1RBCF(I), Q1ABCF(I)                         LIN02770
   40 CONTINUE                                                          LIN02780
   99 FORMAT('0',5X,I2,10X,I2,10X,2(E14.7,5X,E14.7,10X))                LIN02790
   90 FORMAT( 'C',10X,'SUBROUTINE Q1COEF -  COMPLEX POWER SERIES COEFFICLIN02800
     1IENTS')                                                           LIN02810
   94 FORMAT( '0',5X,'INDEX',7X,'DEG FCLY',4X,'REFERENCE BLADE TERMS Q1CLIN02820
     2COF(INDEX)',7X,'ADJACENT BLADE TERMS Q1COF(2*INDEX)')             LIN02830
      RETURN                                                            LIN02840
      END                                                               LIN02850
      SUBROUTINE Q1PPOT(DK,DR,DW,RHO,OFFSET,SIGMA,N,X,Q1ABCF,Q1RBCF,IPT)LIN02860
      IMPLICIT REAL * 8(A-H,O,P,R-Y), COMPLEX * 16(Z,Q)                 LIN02870
      DIMENSION Q1ABCF(13), Q1RBCF(13), GRR(13), GAA(13), GPHI(13)      LIN02880
```

```
      DIMENSION QDRR(13),QDAA(13),QDPHI(13)                                  LIN02890
      DIMENSION X(13)                                                        LIN02900
      IF(IPT.LE.0) RETURN                                                    LIN02910
      WRITE (6,990) DK,DR,DW,RHO,OFFSET,SIGMA,N,IPT                          LIN02920
  990 FORMAT('1',10X,'Q1PPOT -CALCULATION OF COMPLEX POTENTIALS FOR ZONE'LIN02930
     1 ONE',/,'0',10X,'THESE CALCULATIONS ARE BASED ON THE SPANNING FUNCLIN02940
     2TION COEFFICIENTS GIVEN DIRECTLY ABOVE',/                          LIN02950
     3 ,' ',10X,'DK',13X,'DR',13X,'DW',13X,'RHO',12X,'OFFSET',9X,'SIGMA'LIN02960
     4 ,10X,'N',3X,'IPT',/,' ',10X,                                      LIN02970
     5(E12.5,3X),I2,2X,I3)                                               LIN02980
   90 FORMAT ('0 X-STATION NUMBER ',I2,', XSTN = ',F6.4,/,                LIN02990
     1'   BL TOTAL POTENTIAL',14X,'REF BL POTENTIAL',8X,                 LIN03000
     2 'ADJ BL POTENTIAL',8X,'REF BL INT POTENTIAL',5X,                  LIN03010
     3 'ADJ BL INT POTENTIAL')                                           LIN03020
   91 FORMAT('  R  ',2(E14.7,','),8(E11.4,','))                         LIN03030
   92 FORMAT('  A  ',2(E14.7,','),8(E11.4,','))                         LIN03040
      ICT = IPT - 1                                                       LIN03050
      RZERO = 0.0D0                                                       LIN03060
      DO 10 I = 1,N                                                       LIN03070
      XSTN = X(I)                                                         LIN03080
      QRO = Q1PRBP(DK,DR,DW,RZERO,XSTN,IOT)                               LIN03090
      QRP = Q1PRBP(DK,DR,DW,RHO,XSTN,ICT)                                 LIN03100
      QAO = Q1PABP(DK,DR,DW,RZERO,OFFSET,SIGMA,XSTN,IOT)                  LIN03110
      QAP = Q1PABP(DK,DR,DW,RHO,OFFSET,SIGMA,XSTN,ICT)                    LIN03120
      QRIPO = Q1PRIP(DK,DR,DW,RZERO,OFFSET,XSTN,C1RBCF,N,IOT)             LIN03130
      QRIPP = Q1PRIP(DK,DR,DW,RHO,OFFSET,XSTN,Q1RBCF,N,ICT)               LIN03140
      QAIPO = Q1PAIP(DK,DR,DW,RZERO,OFFSET,SIGMA,XSTN,Q1ABCF,N,ICT)       LIN03150
      QAIPP = C1PAIP(DK,DR,DW,RHO,OFFSET,SIGMA,XSTN,C1ABCF,N,ICT)         LIN03160
      QR = QRO + QAP + QRIPO + QAIPP                                      LIN03170
      QA = QRP + QAO + QRIPP + QAIPO                                      LIN03180
      QPHI(I)=Q1PPHI(DK,DR,DW,RHO,OFFSET,SIGMA,N,Q1ABCF,Q1RBCF,XSTN,IOT)  LIN03190
      WRITE (6,90) I,XSTN                                                 LIN03200
      WRITE(6,91) QR,QRO,QAP,QRIPO,QAIPP                                  LIN03210
      WRITE(6,92) QA,QRP,QAO,QRIPP,QAIPO                                  LIN03220
      QRR(I) = QR                                                         LIN03230
      QAA(I) = QA                                                         LIN03240
      QDRO = Q1DRBP(DK,DR,DW,RZERO,XSTN,IOT)                              LIN03250
      QDRP = Q1DRBP(DK,DR,DW,RHO,XSTN,ICT)                                LIN03260
      QDAO = Q1DABP(DK,DR,DW,RZERO,OFFSET,SIGMA,XSTN,IOT)                 LIN03270
      QDAP = Q1DABP(DK,DR,DW,RHO,OFFSET,SIGMA,XSTN,ICT)                   LIN03280
      QDRIPO = Q1DRIP(DK,DR,DW,RZERO,OFFSET,XSTN,C1RBCF,N,ICT)            LIN03290
      QDRIPP = Q1DRIP(DK,DR,DW,RHO,OFFSET,XSTN,Q1RBCF,N,IOT)              LIN03300
      QDAIPO = Q1DAIP(DK,DR,DW,RZERO,OFFSET,SIGMA,XSTN,C1ABCF,N,IOT)      LIN03310
      QDAIPP = Q1DAIP(DK,DR,DW,RHO,OFFSET,SIGMA,XSTN,Q1ABCF,N,IOT)        LIN03320
      QDR = QDRO + QDAP + QDRIPO + QDAIPP                                 LIN03330
      QDA = QDRP + QDAO + QDRIPP + QDAIPO                                 LIN03340
      QDPHI(I)=Q1DPHI(DK,DR,DW,RHO,OFFSET,SIGMA,N,C1ABCF,C1RBCF,          LIN03350
     1     XSTN,ICT)                                                      LIN03360
```

```
      WRITE (6,95) I,XSTN                                                       LIN03370
      WRITE(6,91) CDR,QCRO,CDAP,QDRIPO,CCAIPP                                   LIN03380
      WRITE(6,92) CCA,QCRP,CDAO,QDRIPP,CCAIPO                                   LIN03390
      CCRR(I) = CDR                                                             LIN03400
      QCAA(I) = QCA                                                             LIN03410
   95 FORMAT ('0 X-STATION NUMBER ',I2,', XSTN = ',F6.4,/,                      LIN03420
     1'  BL TOTAL D(PCT)/DX',14X,'REF EL D(POT)/DX',8X,                        LIN03430
     2 'ADJ BL D(POT)/DX',8X,'REF BL INT D(POT)/DX',5X,                         LIN03440
     3 'ADJ BL INT D(POT)/DX')                                                  LIN03450
   10 CONTINUE                                                                  LIN03460
      WRITE(6,994)                                                              LIN03470
  994 FORMAT ('1',10X,'SUMMARY LISTING',/,                                      LIN03480
     1'0',10X,'XSTN',7X,'SINGLE BLADE TOTAL POTENTIAL',6X,                      LIN03490
     1  'REF BLADE POTENTIAL',15X,'ADJ BLADE POTENTIAL')                        LIN03500
      DO 20 I = 1,N                                                             LIN03510
      XSTN = X(I)                                                               LIN03520
      WRITE(6,94) XSTN,CPHI(I),QRR(I),CAA(I)                                    LIN03530
   94 FORMAT(' ',10X,F6.4,3(5X,E14.7,',',E14.7))                                LIN03540
   20 CONTINUE                                                                  LIN03550
      WRITE(6,996)                                                              LIN03560
  996 FORMAT('0',10X,'XSTN',7X,'SINGLE BLADE TOTAL D(POT)/DX',6X,               LIN03570
     1  'REF BLADE D(POT)/DX',15X,'ADJ BLADE D(POT)/DX')                        LIN03580
      DO 50 I = 1,N                                                             LIN03590
      XSTN = X(I)                                                               LIN03600
      WRITE(6,94) XSTN,CDPHI(I),QCRR(I),CDAA(I)                                 LIN03610
   50 CONTINUE                                                                  LIN03620
      RETURN                                                                    LIN03630
      END                                                                       LIN03640
      SUBROUTINE C1DCOF(DK,DR,DW,RHO,OFFSET,SIGMA,N,C1ABCF,C1RBCF,IPT)          LIN03650
      IMPLICIT REAL * 8 (A-H,O,P,R-Y), COMPLEX * 16 (Q,Z)                       LIN03660
      DIMENSION Q1ABCF(13), Q1RBCF(13)                                          LIN03670
      IF(IPT.GT.0)WRITE (6,990) DK,DR,DW,RHO,OFFSET,SIGMA,N,IPT                 LIN03680
  990 FORMAT('1',10X,'Q1DCOF - CALCULATION OF COMPLEX DIMENSIONLESS AERO        LIN03690
     1DYNAMIC COEFFICIENTS',/,                                                  LIN03700
     3 '0',10X,'DK',13X,'DR',13X,'DW',13X,'RHO',12X,'OFFSET',9X,'SIGMA'         LIN03710
     4 ,10X,'N',3X,'IPT',/,' ',10X,                                             LIN03720
     5(E12.5,3X),I2,2X,I3)                                                      LIN03730
      ICT = IPT - 3                                                             LIN03740
  999 CDCL = C1DCL(DK,DR,DW,RHO,OFFSET,SIGMA,N,Q1ABCF,Q1RBCF,ICT)               LIN03750
      CDCM = Q1DCM(DK,DR,DW,RHO,OFFSET,SIGMA,N,C1ABCF,Q1RBCF,ICT)               LIN03760
      GAMMA = 1.400                                                             LIN03770
      TAU = (2.0D0*DSQRT((GAMMA + 1.0D0)*DW))/DR                                LIN03780
      WRITE (6,90) DK,TAU,DW,N,SIGMA,CDCL,CDCM                                  LIN03790
   90 FORMAT('0',5X,'DK = ',F6.3,', TAU = ',F7.4,', DW = ',F6.3,', N = '        LIN03800
     1     ,I2,', SIGMA = ',F6.3,', CL = ',F9.4,',',F9.4,', CM = ',            LIN03810
     2     F9.4,',',F9.4)                                                       LIN03820
      RETURN                                                                    LIN03830
      END                                                                       LIN03840
```

```
      COMPLEX FUNCTION C1DCL*16(DK,DR,DW,RHO,OFFSET,SIGMA,N,Q1ABCF,C1RBC  LIN03850
     1F,IPT)                                                            LIN03860
      IMPLICIT REAL*8(A - E,G,H,O,P,R - Y), COMPLEX*16(F,Q,Z)           LIN03870
      DIMENSION Q1ABCF(13), Q1RBCF(13)                                  LIN03880
      DIMENSION FV(5),F1T(60),F2T(60),F3T(60),CESTT(60),CPSUM(60)       LIN03890
      DIMENSION DAT(60),  ARESTT(60),EPST(60)                           LIN03900
      DIMENSION LCRR(60)                                                LIN03910
      F(X) = (Q1DPHI(DK,DR,DW,RHO,OFFSET,SIGMA,N,C1ABCF,C1RBCF,X,ICT)   LIN03920
     1   -QALPHA*C1PHI(DK,DR,DW,RHO,OFFSET,SIGMA,N,C1ABCF,C1RBCF,X,ICT))LIN03930
      IF (IPT.GT.0) WRITE (6,990) DK,DR,DW,RHO,OFFSET,SIGMA,N,IPT       LIN03940
  990 FORMAT('0',10X,'Q1DCL ENTERED WITH:',/,                           LIN03950
     3  ' ',10X,'DK',13X,'DR',13X,'DK',13X,'RHO',12X,'OFFSET',9X,'SIGMA' LIN03960
     4  ,10X,'N',3X,'IPT',/,' ',10X,                                   LIN03970
     56(E12.5,3X),I2,2X,I3)                                             LIN03980
      A = - 1.0D0                                                       LIN03990
      B = 1.0D0                                                         LIN04000
      ICT = IPT - 1                                                     LIN04010
      GAMMA = 1.4D0                                                     LIN04020
      DLAMDA = DK/((GAMMA + 1.0D0) * DW)                                LIN04030
      QALPHA = DCMPLX(0.0D0,DLAMDA-DK)                                  LIN04040
      U   =  9.0E - 13                                                  LIN04050
      ACC = 1.0D-5                                                      LIN04060
      EPSURU = 4.0*U                                                    LIN04070
      IFLAG = 1                                                         LIN04080
      EPS = ACC                                                         LIN04090
      QERROR = DCMPLX(0.0D0,0.0D0)                                      LIN04100
      LVL = 1                                                           LIN04110
      LCRR(LVL) = 1                                                     LIN04120
      CPSUM(LVL) = 0.0                                                  LIN04130
      ALPHA = A                                                         LIN04140
      DA = B - A                                                        LIN04150
      AREA = 0.0                                                        LIN04160
      AREST = 0.0                                                       LIN04170
      FV(1) = F(ALPHA)                                                  LIN04180
      FV(3) = F(ALPHA + 0.5*DA)                                         LIN04190
      FV(5) = F(ALPHA + DA)                                             LIN04200
      KOUNT = 3                                                         LIN04210
      WT = DA/6.0                                                       LIN04220
      CEST = WT*(FV(1) + 4.0*FV(3) + FV(5))                             LIN04230
    1 DX = 0.5*DA                                                       LIN04240
      FV(2) = F(ALPHA + 0.5*DX)                                         LIN04250
      FV(4) = F(ALPHA + 1.5*DX)                                         LIN04260
      KOUNT = KOUNT + 2                                                 LIN04270
      WT = DX/6.0                                                       LIN04280
      CESTL = WT*(FV(1) + 4.0*FV(2) + FV(3))                            LIN04290
      CESTR = WT*(FV(3) + 4.0*FV(4) + FV(5))                            LIN04300
      CSUM = CESTL + CESTR                                              LIN04310
      ARESTL = WT*(CDABS(FV(1)) + CDABS(4.0*FV(2)) + CDABS(FV(3)))      LIN04320
```

```
      ARESTR = WT*(CDABS(FV(3)) + CDABS(4.0*FV(4)) + CDABS(FV(5)))          LIN04330
      AREA = AREA + ((ARESTL + ARESTR) - AREST)                             LIN04340
      QDIFF = QEST - QSUM                                                   LIN04350
      IF(CDABS(QDIFF).LE.EPS*DABS(AREA))GO TO 2                             LIN04360
      IF(DABS(DX).LE.EFCURU*DABS(ALPHA))GO TO 5                             LIN04370
      IF(LVL.GE.60)GO TO 5                                                  LIN04380
      IF(KCUNT.GE.4000)GO TO 6                                              LIN04390
      LVL = LVL + 1                                                         LIN04400
      LCRR(LVL) = 0                                                         LIN04410
      F1T(LVL) = FV(3)                                                      LIN04420
      F2T(LVL) = FV(4)                                                      LIN04430
      F3T(LVL) = FV(5)                                                      LIN04440
      DA = DX                                                               LIN04450
      DAT(LVL) = DX                                                         LIN04460
      AREST = ARESTL                                                        LIN04470
      ARESTT(LVL) = ARESTR                                                  LIN04480
      QEST = QESTL                                                          LIN04490
      QESTT(LVL) = QESTR                                                    LIN04500
      EPS = EPS/1.4                                                         LIN04510
      EPST(LVL) = EPS                                                       LIN04520
      FV(5) = FV(3)                                                         LIN04530
      FV(3) = FV(2)                                                         LIN04540
      GO TO 1                                                               LIN04550
    2 CERROR = QERROR + QDIFF/15.0                                          LIN04560
    3 IF(LORR(LVL).EQ.0)GO TO 4                                             LIN04570
      QSUM = QPSUM(LVL) + QSUM                                              LIN04580
      LVL = LVL - 1                                                         LIN04590
      IF(LVL.GT.1)GO TO 3                                                   LIN04600
      Q1DCL = QSUM * 2.0D0                                                  LIN04610
      IF(IPT.GT.0) GO TO 11                                                 LIN04620
      IF(IFLAG.EQ.1) RETURN                                                 LIN04630
      WRITE(6,990) DK,DR,DW,RHO,OFFSET,SIGMA,N,IPT                          LIN04640
   11 WRITE(6,995) Q1DCL,IFLAG,CERROR                                       LIN04650
  990 FORMAT('1',15X,'RESULTS: Q1DCL = ',E14.7,',',E14.7,/,                 LIN04660
     1' ',15X,'IFLAG',2X,'QERROR',/,                                        LIN04670
     2' ',15X,I3,2X,E14.7,',',E14.7)                                        LIN04680
      RETURN                                                                LIN04690
    4 QPSUM(LVL) = QSUM                                                     LIN04700
      LCRR(LVL) = 1                                                         LIN04710
      ALPHA = ALPHA + DA                                                    LIN04720
      DA = DAT(LVL)                                                         LIN04730
      FV(1) = F1T(LVL)                                                      LIN04740
      FV(3) = F2T(LVL)                                                      LIN04750
      FV(5) = F3T(LVL)                                                      LIN04760
      AREST = ARESTT(LVL)                                                   LIN04770
      QEST = QESTT(LVL)                                                     LIN04780
      EPS = EPST(LVL)                                                       LIN04790
      GO TO 1                                                               LIN04800
```

```
5    IFLAG = 2                                                                  LINC481C
     GC TO 2                                                                    LINO4820
6    IFLAG = 3                                                                  LINO4830
     GC TO 2                                                                    LINO4E4C
     END                                                                        LINO4850
     CCMPLEX FUNCTION C1DCM*16(DK,DR,DW,RHO,OFFSET,SIGMA,N,Q1ABCF,C1RBCLINO4860
    1F,IPT)                                                                     LINC4870
     IMPLICIT REAL*8(A - E,G,H,O,P,R - Y), COMPLEX*16(F,C,Z)                    LINO4E8C
     DIMENSION Q1ABCF(13), Q1RBCF(13)                                           LINO4890
     DIMENSION FV(5),F1T(60),F2T(60),F3T(60),CESTT(60),CPSUM(60)                LINC49CC
     DIMENSION DAT(60),  ARESTT(60),EPST(60)                                    LINO4910
     DIMENSION LCRR(60)                                                         LINC492C
     F(X) = X*(Q1DPHI(DK,DR,DW,RHO,OFFSET,SIGMA,N,C1ABCF,C1RBCF,X,ICT)          LINO4930
    1   -QALPHA*C1PHI(DK,DR,DW,RHO,OFFSET,SIGMA,N,C1ABCF,C1RBCF,X,ICT))LINO4940
     IF (IPT.GT.0) WRITE (6,990) DK,DR,DW,RHO,OFFSET,SIGMA,N,IPT                LINC495C
990  FORMAT('0',10X,'Q1DCM ENTERED WITH:',/,                                    LINO4960
    3  ' ',10X,'DK',13X,'DR',13X,'DK',13X,'RHO',12X,'OFFSET',9X,'SIGMA'        LINC497C
    4  ,10X,'N',3X,'IPT',/,' ',10X,                                            LINO4980
    56(E12.5,3X),I2,2X,I3)                                                      LINO4990
     A = - 1.0D0                                                                LINO5000
     B = 1.0D0                                                                  LINO5010
     ICT = IPT - 1                                                              LINO5020
     GAMMA = 1.4D0                                                              LINO5C30
     DLAMDA = DK/((GAMMA + 1.0D0) * DW)                                         LINO5040
     QALPHA = DCMPLX(0.0D0,DLAMDA-DK)                                           LINO5050
     U   =  9.0E - 13                                                           LINC5C6C
     ACC = 1.0D-5                                                               LINO5070
     EFGURU = 4.C*U                                                             LINC5C8C
     IFLAG = 1                                                                  LINO5C90
     EPS = ACC                                                                  LINO5100
     CERROR = CCMPLX(0.0DC,0.0D0)                                               LINO5110
     LVL = 1                                                                    LINO5120
     LCRR(LVL) = 1                                                              LINO5130
     CPSUM(LVL) = 0.0                                                           LINO5140
     ALPHA = A                                                                  LINO5150
     DA = B - A                                                                 LINO5160
     AREA = 0.0                                                                 LINC517C
     AREST = 0.0                                                                LINO5180
     FV(1) = F(ALPHA)                                                           LINO5190
     FV(3) = F(ALPHA + 0.5*DA)                                                  LINO52CC
     FV(5) = F(ALPHA + DA)                                                      LINO5210
     KCUNT = 3                                                                  LINC5220
     WT = DA/6.0                                                                LINO5230
     QEST = WT*(FV(1) + 4.0*FV(3) + FV(5))                                      LINO5240
1    DX = 0.5*DA                                                                LINO5250
     FV(2) = F(ALPHA + 0.5*DX)                                                  LINO5260
     FV(4) = F(ALPHA + 1.5*DX)                                                  LINO5270
     KCUNT = KOUNT + 2                                                          LINO5280
```

```
      WT = DX/6.0                                                          LIN05290
      QESTL = WT*(FV(1) + 4.0*FV(2) + FV(3))                               LIN05300
      QESTR = WT*(FV(3) + 4.0*FV(4) + FV(5))                               LIN05310
      QSUM = QESTL + QESTR                                                 LIN05320
      ARESTL = WT*(CDABS(FV(1)) + CDABS(4.0*FV(2)) + CDABS(FV(3)))         LIN05330
      ARESTR = WT*(CDABS(FV(3)) + CDABS(4.0*FV(4)) + CDABS(FV(5)))         LIN05340
      AREA = AREA + ((ARESTL + ARESTR) - AREST)                            LIN05350
      QDIFF = QEST - QSUM                                                  LIN05360
      IF(CDABS(QDIFF).LE.EPS*DABS(AREA))GO TO 2                            LIN05370
      IF(DABS(DX).LE.EPSILRU*DABS(ALPHA))GO TO 5                           LIN05380
      IF(LVL.GE.60)GO TO 5                                                 LIN05390
      IF(KOUNT.GE.4000)GO TO 6                                             LIN05400
      LVL = LVL + 1                                                        LIN05410
      LORR(LVL) = 0                                                        LIN05420
      F1T(LVL) = FV(3)                                                     LIN05430
      F2T(LVL) = FV(4)                                                     LIN05440
      F3T(LVL) = FV(5)                                                     LIN05450
      DA = DX                                                              LIN05460
      DAT(LVL) = DX                                                        LIN05470
      AREST = ARESTL                                                       LIN05480
      ARESTT(LVL) = ARESTR                                                 LIN05490
      QEST = QESTL                                                         LIN05500
      QESTT(LVL) = QESTR                                                   LIN05510
      EPS = EPS/1.4                                                        LIN05520
      EPST(LVL) = EPS                                                      LIN05530
      FV(5) = FV(3)                                                        LIN05540
      FV(3) = FV(2)                                                        LIN05550
      GO TO 1                                                              LIN05560
    2 QERROR = QERROR + QDIFF/15.0                                         LIN05570
    3 IF(LORR(LVL).EQ.0)GO TO 4                                            LIN05580
      QSUM = QPSUM(LVL) + QSUM                                             LIN05590
      LVL = LVL - 1                                                        LIN05600
      IF(LVL.GT.1)GO TO 3                                                  LIN05610
      Q1DCM = QSUM * 2.0D0                                                 LIN05620
      IF(IPT.GT.0) GO TO 11                                                LIN05630
      IF(IFLAG.EQ.1) RETURN                                                LIN05640
      WRITE(6,990) DK,DR,DW,RHO,OFFSET,SIGMA,N,IPT                         LIN05650
   11 WRITE(6,995) Q1DCM,IFLAG,QERROR                                      LIN05660
  995 FORMAT(' ',15X,'RESULTS: Q1DCM = ',E14.7,',',E14.7,/,                LIN05670
     1' ',15X,'IFLAG',2X,'QERROR',/,                                       LIN05680
     2' ',15X,I3,2X,E14.7,',',E14.7)                                       LIN05690
      RETURN                                                               LIN05700
    4 QPSUM(LVL) = QSUM                                                    LIN05710
      LORR(LVL) = 1                                                        LIN05720
      ALPHA = ALPHA + DA                                                   LIN05730
      DA = DAT(LVL)                                                        LIN05740
      FV(1) = F1T(LVL)                                                     LIN05750
      FV(3) = F2T(LVL)                                                     LIN05760
```

```
      FV(5) = F3T(LVL)                                                  LIN05770
      AREST = ARESTT(LVL)                                               LIN05780
      QEST = QESTT(LVL)                                                 LIN05790
      EPS = EPST(LVL)                                                   LIN05800
      GO TO 1                                                           LIN05810
    5 IFLAG = 2                                                         LIN05820
      GO TO 2                                                           LIN05830
    6 IFLAG = 3                                                         LIN05840
      GO TO 2                                                           LIN05850
      END                                                               LIN05860
      COMPLEX FUNCTION C1PHI*16 (DK,DR,DW,RHO,OFFSET,SIGMA,             LIN05870
     1    N,Q1ABCF,Q1RBCF,XSTN,IPT)                                     LIN05880
      IMPLICIT REAL*8(A-H,O,P,R-Y), COMPLEX*16(Q,Z)                     LIN05890
      DIMENSION Q1ABCF(13), Q1RBCF(13)                                  LIN05900
      IF (IPT.GT.0) WRITE(6,990) DK,DR,DW,RHO,OFFSET,SIGMA,XSTN,IPT     LIN05910
  990 FORMAT('0',10X,'Q1PHI ENTERED WITH:',/,                           LIN05920
     1   ' ',10X,'DK',11X,'DR',11X,'DW',11X,'RHO',10X,'OFFSET',7X,'SIGMA'LIN05930
     2   ,8X,'XSTN',9X,'IPT'/,' ',10X,7(E12.5,','),I3)                 LIN05940
      RZERO = 0.0D0                                                     LIN05950
      IOT = IPT - 1                                                     LIN05960
      C1PHI = Q1PREP(DK,DR,DW,RZERO,XSTN,IOT)                           LIN05970
     1 + Q1PRIP(DK,DR,DW,RZERO,OFFSET,XSTN,Q1RBCF,N,IOT)                LIN05980
     2 + C1PAEP(DK,DR,DW,DR,OFFSET,SIGMA,XSTN,IOT)                     LIN05990
     3 + Q1PAIP(DK,DR,DW,DR,OFFSET,SIGMA,XSTN,Q1ABCF,N,IOT)             LIN06000
      IF(IPT.LE.0) RETURN                                               LIN06010
      WRITE (6,995) Q1PHI                                               LIN06020
  995 FORMAT('0',10X,'C1PHI = ',E14.7,',',E14.7)                        LIN06030
      RETURN                                                            LIN06040
      END                                                               LIN06050
      COMPLEX FUNCTION Q1PRBP*16(DK,DR,DW,RHO,XSTN,IPT)                 LIN06060
      IMPLICIT REAL * 8 (A-H,O,P,R-Y), COMPLEX * 16 (Q,Z)               LIN06070
      DIMENSION QINP(2)                                                 LIN06080
      IF (IPT.GT.0) WRITE(6,990) DK,DR,DW,RHO,XSTN,IPT                  LIN06090
  990 FORMAT('0',10X,'C1PRBP ENTERED WITH:',/,                          LIN06100
     1   ' ',10X,'DK',11X,'DR',11X,'DW',11X,'RHO',10X,'XSTN',9X,'IPT'   LIN06110
     2   ,/,' ',10X,5(E12.5,','),I3)                                   LIN06120
      IF(XSTN.LE.RHO-1.0D0) GOTO 20                                     LIN06130
      IOT = IPT - 1                                                     LIN06140
      GAMMA = 1.4D0                                                     LIN06150
      DM2 = (GAMMA + 1.0D0) * DW                                        LIN06160
      DLAMDA = DK/DM2                                                   LIN06170
      PR = 1.0+XSTN-RHO                                                 LIN06180
      PIMAG = (DK+DLAMDA)*((XSTN-RHO)*(XSTN-RHO) - 1.0D0)               LIN06190
      Q1PRBP = - DCMPLX(PR,PIMAG-DLAMDA*XSTN*PR)/DSQRT(DM2)             LIN06200
      IF(IPT.LE.0) RETURN                                               LIN06210
      GOTO 60                                                           LIN06220
   20 C1PREP = DCMPLX(0.0D0,0.0D0)                                      LIN06230
      IF(IPT.LE.0) RETURN                                               LIN06240
```

```
 60 WRITE (6,995) C1PRBP                                                LIN06250
995 FORMAT('0',10X,'Q1PRBP = ',E14.7,',',E14.7)                         LIN06260
    RETURN                                                              LIN06270
    END                                                                 LIN06280
    COMPLEX FUNCTION Q1PAEP*16(DK,DR,DW,RHO,OFFSET,SIGMA,XSTN,IPT)      LIN06290
    IMPLICIT REAL * 8 (A-H,O,P,R-Y), COMPLEX * 16 (Q,Z)                 LIN06300
    DIMENSION QINP(2)                                                   LIN06310
    IF(IPT.GT.0) WRITE(6,990) DK,DR,DW,RHO,OFFSET,SIGMA,XSTN,IPT        LIN06320
990 FORMAT('0',10X,'Q1PAEP ENTERED WITH:',/,                            LIN06330
   1  ' ',10X,'DK',11X,'DR',11X,'DW',11X,'RHO',10X,'OFFSET',7X,        LIN06340
   2  'SIGMA',8X,'XSTN',9X,'IPT'/,' ',10X,7(E12.5,','),I3)             LIN06350
                                                                        LIN06360
    XASTN = XSTN - OFFSET                                               LIN06370
    IF(XASTN.LE.RHO-1.0D0) GOTO 20                                      LIN06380
    ICT = IPT - 1                                                       LIN06390
    CCONST = CDEXP( DCMPLX(0.0D0,SIGMA))                               LIN06400
    GAMMA = 1.4D0                                                       LIN06410
    DM2 = (GAMMA + 1.0D0) * DW                                          LIN06420
    DLAMDA = DK/DM2                                                     LIN06430
    PR = 1.0D0 + XASTN - RHO                                            LIN06440
    PIMAG = (DK+DLAMDA)*((XASTN-RHO)*(XASTN-RHO) - 1.0D0)               LIN06450
    C1PAEP =  DCMPLX(PR,PIMAG-DLAMDA*XASTN*PR)*(CCONST/DSQRT(DM2))      LIN06460
    IF(IPT.LE.0) RETURN                                                 LIN06470
    GO TO 60                                                            LIN06480
 20 C1PAEP = DCMPLX(0.0D0,0.0D0)                                        LIN06490
    IF (IPT.LE.0) RETURN                                                LIN06500
 60 WRITE (6,995) Q1PABP                                                LIN06510
995 FORMAT('0',10X,'Q1PABP = ',E14.7,',',E14.7)                         LIN06520
    RETURN                                                              LIN06530
    END                                                                 LIN06540
    COMPLEX FUNCTION C1DPHI*16 (DK,DR,DW,RHO,OFFSET,SIGMA,              LIN06550
   1     N,Q1ABCF,Q1RBCF,XSTN,IPT)                                      LIN06560
    IMPLICIT REAL*8(A-H,O,P,R-Y), COMPLEX*16(C,Z)                       LIN06570
    DIMENSION Q1ABCF(13),Q1RBCF(13)                                     LIN06580
    IF (IPT.GT.0) WRITE(6,990) DK,DR,DW,RHO,OFFSET,SIGMA,XSTN,IPT       LIN06590
990 FORMAT('0',10X,'Q1DPHI ENTERED WITH:',/,                            LIN06600
   1  ' ',10X,'DK',11X,'DR',11X,'DW',11X,'RHO',10X,'OFFSET',7X,'SIGMA'LIN06610
   2  ,8X,'XSTN',9X,'IPT'/,' ',10X,7(E12.5,','),I3)                    LIN06620
    RZERO = 0.0D0                                                       LIN06630
    ICT = IPT - 1                                                       LIN06640
    C1DPHI = Q1DRBP(DK,DR,DW,RZERO,XSTN,IOT)                            LIN06650
   1 + Q1DRIP(DK,DR,DW,RZERO,OFFSET,XSTN,Q1RBCF,N,IOT)                  LIN06660
   2 + C1DAEP(DK,DR,DW,DR,OFFSET,SIGMA,XSTN,IOT)                        LIN06670
   3 + Q1DAIP(DK,DR,DW,DR,OFFSET,SIGMA,XSTN,Q1AECF,N,IOT)               LIN06680
    IF(IPT.LE.0) RETURN                                                 LIN06690
    WRITE (6,995) Q1DPHI                                                LIN06700
995 FORMAT('0',10X,'C1DPHI = ',E14.7,',',E14.7)                         LIN06710
    RETURN                                                              LIN06720
    END
```

```
      COMPLEX FUNCTION Q1DREP*16(DK,DR,DW,RHO,XSTN,IPT)                 LIN06730
      IMPLICIT REAL * 8 (A-H,O,P,R-Y), COMPLEX * 16 (Q,Z)               LIN06740
      DIMENSION CINP(2)                                                 LIN06750
      IF (IPT.GT.0) WRITE(6,990) DK,DR,DW,RHO,XSTN,IPT                  LIN06760
  990 FORMAT('0',10X,'Q1DRBP ENTERED WITH:',/,                          LIN06770
     1   ' ',10X,'DK',11X,'DR',11X,'DW',11X,'RHO',10X,'XSTN',9X,'IPT'  LIN06780
     2  ,/,' ',10X,5(E12.5,','),I3)                                    LIN06790
      IF(XSTN.LE.RHO-1.0D0) GOTO 20                                     LIN06800
      ICT = IPT - 1                                                     LIN06810
      GAMMA = 1.4D0                                                     LIN06820
      DM2 = (GAMMA + 1.0D0) * DW                                        LIN06830
      DLAMDA = DK/DM2                                                   LIN06840
      DM2 = (GAMMA + 1.0D0) * DW                                        LIN06850
      PIMAG = (DK+DLAMDA)*(XSTN-RHO)                                    LIN06860
      Q1DRBP = -DCMPLX(1.0D0,PIMAG-DLAMDA*XSTN)/DSQRT(DM2)              LIN06870
      IF(IPT.LE.0) RETURN                                               LIN06880
      GOTO 60                                                           LIN06890
   20 Q1DRBP = DCMPLX(0.0D0,0.0D0)                                      LIN06900
      IF(IPT.LE.0) RETURN                                               LIN06910
   60 WRITE (6,995) Q1DRBP                                              LIN06920
  995 FORMAT('0',10X,'Q1DRBP = ',E14.7,',',E14.7)                       LIN06930
      RETURN                                                            LIN06940
      END                                                               LIN06950
      COMPLEX FUNCTION Q1DABP*16(DK,DR,DW,RHO,OFFSET,SIGMA,XSTN,IPT)    LIN06960
      IMPLICIT REAL * 8 (A-H,O,P,R-Y), COMPLEX * 16 (Q,Z)               LIN06970
      DIMENSION CINP(2)                                                 LIN06980
      IF(IPT.GT.0) WRITE(6,990) DK,DR,DW,RHO,OFFSET,SIGMA,XSTN,IPT      LIN06990
  990 FORMAT('0',10X,'Q1DABP ENTERED WITH:',/,                          LIN07000
     1   ' ',10X,'DK',11X,'DR',11X,'DW',11X,'RHO',10X,'OFFSET',7X,     LIN07010
     2  'SIGMA',8X,'XSTN',9X,'IPT'/,' ',10X,7(E12.5,','),I3)           LIN07020
      XASTN = XSTN - OFFSET                                             LIN07030
      IF(XASTN.LE.RHO-1.0D0) GOTO 20                                    LIN07040
      ICT = IPT - 1                                                     LIN07050
      CCONST = CDEXP( DCMPLX(0.0D0,SIGMA))                              LIN07060
      GAMMA = 1.4D0                                                     LIN07070
      DM2 = (GAMMA + 1.0D0) * DW                                        LIN07080
      DLAMDA = DK/DM2                                                   LIN07090
      PIMAG = (DK+DLAMDA)*(XASTN-RHO)                                   LIN07100
      Q1DABP =  DCMPLX(1.0D0,PIMAG-DLAMDA*XASTN)*CCONST/DSQRT(DM2)      LIN07110
      IF(IPT.LE.0) RETURN                                               LIN07120
      GO TO 60                                                          LIN07130
   20 Q1DABP = DCMPLX(0.0D0,0.0D0)                                      LIN07140
      IF (IPT.LE.0) RETURN                                              LIN07150
   60 WRITE (6,995) Q1DABP                                              LIN07160
  995 FORMAT('0',10X,'Q1DABP = ',E14.7,',',E14.7)                       LIN07170
      RETURN                                                            LIN07180
      END                                                               LIN07190
      COMPLEX FUNCTION Q1PAIP*16(DK,DR,DW,RHO,OFST,SGMA,XSTN,Q1OF,N,IPT)LIN07200
```

```
      IMPLICIT REAL*8 (A-H,C,P,R-Y), COMPLEX * 16 (C,Z)                  LIN07210
      DIMENSION Q1CF(13)                                                LIN07220
      IF (IPT.GT.0) WRITE(6,990) DK,DR,DW,RHO,OFST,SGMA,XSTN,IPT        LIN07230
  990 FORMAT('0',10X,'Q1PAIP ENTERED WITH:',/,                          LIN07240
     1   ' ',10X,'DK',11X,'DR',11X,'DW',11X,'RHO',10X,'OFST',9X,'SGMA',9XLIN07250
     2  ,'XSTN',9X,'IPT'/,' ',10X,7(E12.5,','),I3)                     LIN07260
      XASTN = XSTN - OFST                                               LIN07270
      IF(XASTN.LE.DR+RHO-1.0D0) GO TO 20                                LIN07280
      CCONST = CDEXP( DCMPLX(0.0D0,SGMA))                               LIN07290
      GAMMA = 1.4DO                                                     LIN07300
      DM2 = (GAMMA+1.0DO)*DW                                            LIN07310
      DLAMDA = DK/DM2                                                   LIN07320
      XX=XASTN-RHO                                                      LIN07330
      Q1 = Q1CF(1)                                                      LIN07340
      Q2 = Q1CF(2)                                                      LIN07350
      PR1 = XX+1.CDO-DR                                                 LIN07360
      PR2 = (DR-1.0DO - XX/2.0DO)*XX - (DR-1.0DO)*(DR-1.CDC)/2.0DO      LIN07370
      PIMAG1 = (XX*XX+(2.0DO-DR)*DR-1.CDC)*DLAMDA/2.0DO                 LIN07380
      PIMAG2 =  XX*XX*((DR-1.0DO)/2.0DO-XX)-((DR-1.0DO)**3)/6.0DO       LIN07390
      PIMAG1=PIMAG1*DLAMDA                                              LIN07400
      PIMAG2 =  PIMAG2 *  DLAMDA                                        LIN07410
      QP1 = DCMPLX(PR1,PIMAG1-DLAMDA*PR1*XASTN)                         LIN07420
      QP2 = DCMPLX(PR2,PIMAG2-DLAMDA*PR2*XASTN)                         LIN07430
      Q1PAIP = (Q1*QP1 + Q2*QP2)/DSQRT(DM2)                             LIN07440
      IF(IPT.LE.0) RETURN                                               LIN07450
      GOTO 30                                                           LIN07460
   20 Q1PAIP = DCMPLX(0.0DO,0.0DO)                                      LIN07470
      IF(IPT.LE.0)RETURN                                                LIN07480
   30 WRITE (6,995) Q1PAIP                                              LIN07490
  995 FORMAT('0',10X,'Q1PAIP = ',E14.7,',',E14.7)                       LIN07500
      RETURN                                                            LIN07510
      END                                                               LIN07520
      COMPLEX FUNCTION C1PRIP*16(DK,DR,DW,RHO,OFFSET,XSTN,C1CF,N,IPT)   LIN07530
      IMPLICIT REAL*8 (A-H,O,P,R-Y), COMPLEX * 16 (Q,Z)                 LIN07540
      DIMENSION C1CF(13)                                                LIN07550
      IF (IPT.GT.0) WRITE(6,990) DK,DR,DW,RHO,XSTN,IPT                  LIN07560
  990 FORMAT('C',10X,'Q1PRIP ENTERED WITH:',/,                          LIN07570
     1   ' ',10X,'DK',11X,'DR',11X,'DW',11X,'RHO',10X,                  LIN07580
     2  'XSTN',9X,'IPT'/,' ',10X,5(E12.5,','),I3)                       LIN07590
      IF(XSTN.LE.DR+RHO+OFFSET-1.0D0) GOTO 20                           LIN07600
      GAMMA = 1.4DO                                                     LIN07610
      DM2 = (GAMMA+1.0DO)*DW                                            LIN07620
      DLAMDA = DK/DM2                                                   LIN07630
      XX = XSTN - RHO                                                   LIN07640
      C2 = C1CF(2)                                                      LIN07650
      C1 = C1CF(1)                                                      LIN07660
      PR1 = XX+1.0DO-DR                                                 LIN07670
      PR2 = (DR-1.0DO - XX/2.0DO)*XX - (DR-1.0DO)*(DR-1.CDC)/2.0DO      LIN07680
```

```
      FIMAG1 = (XX*XX+(2.0D0-DR)*DR-1.0D0)*DLAMDA/2.0D0                  LIN07690
      PIMAG2 =  XX*XX*((DR-1.0D0)/2.0D0-XX)-((DR-1.0D0)**3)/6.0D0       LIN07700
      PIMAG1=PIMAG1*DLAMDA                                              LIN07710
      PIMAG2 = PIMAG2 * DLAMDA                                          LIN07720
      CP1 = DCMPLX(PR1,PIMAG1-DLAMDA*PR1*XSTN)                          LIN07730
      QP2 = DCMPLX(PR2,PIMAG2-DLAMDA*PR2*XSTN)                          LIN07740
      C1PRIP =-(Q1*QP1+C2*CP2)/DSQRT(DM2)                               LIN07750
      IF(IPT.LE.0) RETURN                                               LIN07760
      GOTO 30                                                           LIN07770
   20 C1PRIP = DCMPLX(0.0D0,0.0D0)                                      LIN07780
      IF (IPT.LE.0) RETURN                                              LIN07790
   30 WRITE (6,995) Q1PRIP                                              LIN07800
  995 FORMAT('0',10X,'Q1PRIP = ',E14.7,',',E14.7)                       LIN07810
      RETURN                                                            LIN07820
      END                                                               LIN07830
      COMPLEX FUNCTION Q1DAIP*16(DK,DR,DW,RHO,OFST,SGMA,XSTN,C1CF,N,IPT)LIN07840
      IMPLICIT REAL*8 (A-H,O,P,R-Y), COMPLEX * 16 (C,Z)                 LIN07850
      DIMENSION Q1CF(13)                                                LIN07860
      IF (IPT.GT.0) WRITE(6,990) DK,DR,DW,RHO,OFST,SGMA,XSTN,IPT        LIN07870
  990 FORMAT('0',10X,'Q1DAIP ENTERED WITH:',/,                          LIN07880
     1   ' ',10X,'DK',11X,'DR',11X,'DW',11X,'RHO',10X,'OFST',9X,'SGMA',9XLIN07890
     2   ,'XSTN',9X,'IPT'/,' ',10X,7(E12.5,','),I3)                    LIN07900
      XASTN = XSTN - OFST                                               LIN07910
      IF(XASTN.LE.DR+RHO-1.0D0) GO TO 20                                LIN07920
      CCONST = CDEXP( DCMPLX(0.0D0,SGMA))                               LIN07930
      GAMMA = 1.4D0                                                     LIN07940
      DM2 = (GAMMA+1.0D0)*DW                                            LIN07950
      DLAMDA = DK/DM2                                                   LIN07960
      XX = XASTN - RHO                                                  LIN07970
      Q2 = Q1CF(2)                                                      LIN07980
      Q1 = Q1CF(1)                                                      LIN07990
      PR1 = 1.0D0                                                       LIN08000
      PR2 = DR-1.0D0-XX                                                 LIN08010
      PIMAG1 = DLAMDA*XX                                                LIN08020
      PIMAG2 =DLAMDA*(DR-1.0D0-XX)*XX*XX                                LIN08030
      CP1 = DCMPLX(PR1,PIMAG1-DLAMDA*PR1*XASTN)                         LIN08040
      CP2 = DCMPLX(PR2,PIMAG2-DLAMDA*PR2*XASTN)                         LIN08050
      C1DAIP = (Q1*QP1 + Q2*CP2)/DSQRT(DM2)                             LIN08060
      IF(IPT.LE.0) RETURN                                               LIN08070
      GOTO 30                                                           LIN08080
   20 C1DAIP = DCMPLX(0.0D0,0.0D0)                                      LIN08090
      IF(IPT.LE.0)RETURN                                                LIN08100
   30 WRITE (6,995) Q1DAIP                                              LIN08110
  995 FORMAT('0',10X,'Q1DAIP = ',E14.7,',',E14.7)                       LIN08120
      RETURN                                                            LIN08130
      END                                                               LIN08140
      COMPLEX FUNCTION C1DRIP*16(DK,DR,DW,RHO,OFFSET,XSTN,C1CF,N,IPT)   LIN08150
      IMPLICIT REAL*8 (A-H,O,P,R-Y), COMPLEX * 16 (Q,Z)                 LIN08160
```

```
      DIMENSION Q1CF(13)                                                LINO8170
      IF (IPT.GT.0) WRITE(6,990) DK,DR,DW,RHO,XSTN,IPT                  LINO8180
  990 FORMAT('0',10X,'C1DRIP ENTERED WITH:',/,                          LINO8190
     1   ' ',10X,'DK',11X,'DR',11X,'DW',11X,'RHO',10X,                  LINO8200
     2   'XSTN',9X,'IPT'/,' ',10X,5(E12.5,','),I3)                      LINO8210
      IF(XSTN.LE.DR+RHO+OFFSET-1.0D0) GOTO 20                           LINO8220
      GAMMA = 1.4D0                                                     LINO8230
      DM2 = (GAMMA+1.0D0)*DW                                            LINO8240
      DLAMDA = DK/DM2                                                   LINO8250
      Q1 = Q1CF(1)                                                      LINO8260
      Q2 = Q1CF(2)                                                      LINO8270
      XX = XSTN - RHO                                                   LINO8280
      PR1 = 1.0D0                                                       LINO8290
      PR2 = DR-1.0D0-XX                                                 LINO8300
      PIMAG2 =DLAMDA*(DR-1.0D0-XX)*XX*XX                                LINO8310
      PIMAG1 = DLAMDA*XX                                                LINO8320
      QP1=DCMPLX(PR1,PIMAG1-DLAMDA*PR1*XSTN)                            LINO8330
      QP2 = DCMPLX(PR2,PIMAG2-DLAMDA*PR2*XSTN)                          LINO8340
      Q1DRIP =-(Q1*QP1+Q2*QP2)/DSQRT(DM2)                               LINO8350
      IF(IPT.LE.0) RETURN                                               LINO8360
      GOTO 30                                                           LINO8370
   20 Q1DRIP = DCMPLX(0.0D0,0.0D0)                                      LINO8380
      IF (IPT.LE.0) RETURN                                              LINO8390
   30 WRITE (6,995) Q1DRIP                                              LINO8400
  995 FORMAT('0',10X,'Q1DRIP = ',E14.7,',',E14.7)                       LINO8410
      RETURN                                                            LINO8420
      END                                                               LINO8430
```

LIST OF REFERENCES

1. Elder, P. R., A Theoretical Analysis of Unsteady Transonic Cascade Flow, Thesis, Naval Postgraduate School, Monterey, California, 1972.

2. Schlein, P. B., A Study of Unsteady Transonic Interference Effects, Thesis, Naval Postgraduate School, Monterey, California, 1975.

3. Landahl, M., Unsteady Transonic Flow, Pergamon Press, 1961.

4. Gorelov, D. N., "Oscillations of a Plate Cascade in a Transonic Gas Flow," Mekhanika Zhidkosti i Gaza, v. 1, no. 1, pp. 69-74, 1966.

5. Garrick, I. E. and Rubinow, S. J., Flutter and Oscillating Air-Force Calculations for an Airfoil in Two-Dimensional Supersonic Flow, NACA Report 846, 1946.

6. Stevens, W. P.G. J. Meyers, and L. L. Constantine, "Structured Design," IBM Systems Journal, No. 2, pp. 115-139, 1974.

7. Shampine, L. F., and R. C. Allen, Numerical Computing: An Introduction, W. B. Saunders Company, 1973.

8. Ashley, H. and Landahl, M., Aerodynamics of Wings and Bodies, Addison-Wesley Publishing Company, Inc., 1965.

9. Miles, J. W., "The Compressible Flow Past an Oscillating Airfoil in a Wind Tunnel," Journal of Aeronautical Sciences, v. 23, no. 7, pp. 671-678, 1956.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Documentation Center Cameron Station Alexandria, Virginia 22314	2
2.	Library, Code 0142 Naval Postgraduate School Monterey, California 93940	2
3.	Professor M.F. Platzer, Code 57Pl Department of Aeronautics Naval Postgraduate School Monterey, California 93940	10
4.	Lt. Peter C. Olsen USCG 16207 Pointer Ridge Drive Bowie, Md. 20716	4
5.	Assoc. Professor F.R. Richards, Code 55rh Department of Operations Research Naval Postgraduate School Monterey, Ca. 93940	1
6.	Mr. Robert C. Olsen Quality Engineering, WSOB Pratt and Whitney Aircraft East Hartford, Conn. 08106	1
7.	Professor D.J. Collins, Code 67Co Department of Aeronautics Naval Postgraduate School Monterey, Ca. 93940	1

Thesis
0484
c.1

178764

Olsen

Theoretical analysis of transonic flow past unstaggered oscillating cascades.

Thesis
0484
c.1

178764

Olsen

Theoretical analysis of transonic flow past unstaggered oscillating cascades.

thesO484
Theoretical analysis of transonic flow p
3 2768 001 97502 2
DUDLEY KNOX LIBRARY